



**Universität Bielefeld**  
**Technische Fakultät**  
**Angewandte Informatik**

---

Diplomarbeit

# Objektbezogene 3D-Erkennung automatisch generierter Konturmodelle in Intensitätsbildern

von

Marc Alexander Hanheide

Betreuer: Prof. Dr.-Ing. Gerhard Sagerer  
PD Dr.-Ing. Franz Kummert

Bielefeld, 31. Juli 2001

---

---

## Erklärung

Hiermit versichere ich, dass ich diese Arbeit mit dem Titel

**Objektbezogene 3D-Erkennung automatisch generierter  
Konturmodelle in Intensitätsbildern**

selbständig bearbeitet und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie Zitate kenntlich gemacht habe.

Bielefeld, den 31. Juli 2001

Marc Alexander Hanheide

## Danksagung

Für die Unterstützung bei der Erstellung dieser Arbeit möchte ich meinen Betreuern Gerhard Sagerer und Franz Kummert danken. Auch Lars Krüger von DAIMLERCHRYSLER hat mich mit Diskussionen und Anregungen zum Thema dieser Arbeit unterstützt. Natürlich geht auch ein großes Dankeschön an Anika, dafür, dass sie so geduldig mit mir gewesen ist, wenn ich nur noch „modellbasierte Objekterkennung“ im Kopf hatte. Insbesondere den Korrekturlesern und -leserinnen gebührt mein Dank, als da wären: Alexander Rüegg, Birgit Möller und Thomas Plötz, die mir auch beim täglichen Kampf mit C++ und L<sup>A</sup>T<sub>E</sub>X hilfreich zur Seite standen.

---



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Erkennung räumlicher Modelle</b>	<b>3</b>
2.1	Dreidimensionale Klassifikationsverfahren . . . . .	4
2.2	Ansichtsbezogene Verfahren . . . . .	4
2.3	Objektbezogene Verfahren . . . . .	6
2.4	Initiale Lagehypothese . . . . .	8
2.5	Merkmale . . . . .	8
2.6	Distanzmaße . . . . .	9
2.7	Projektionsoptimierung . . . . .	10
2.8	Modellklassifikation . . . . .	11
2.9	Ziele dieser Arbeit . . . . .	11
<b>3</b>	<b>Modellaufbereitung</b>	<b>13</b>
3.1	Ablauf der Modellaufbereitung . . . . .	14
3.2	Modelldaten . . . . .	15
3.3	Auswahl von Identifikationsmerkmalen . . . . .	17
3.4	Sichtbarkeitsregeln . . . . .	18

<b>4 Objekterkennung</b>	<b>23</b>
4.1 Bildmerkmale . . . . .	23
4.2 Kanten-Distanz-Bild . . . . .	25
4.3 Projektion . . . . .	28
4.4 Optimierung . . . . .	30
4.5 Klassifikation . . . . .	35
4.6 Zusammenfassung . . . . .	35
<b>5 Realisierung</b>	<b>37</b>
5.1 Anforderungen . . . . .	37
5.2 Software-Architektur . . . . .	38
<b>6 Auswertung</b>	<b>41</b>
6.1 Automatische Modellvorbereitung . . . . .	41
6.2 Erkennung . . . . .	43
<b>7 Zusammenfassung &amp; Ausblick</b>	<b>49</b>
<b>A Einführung in die räumliche Geometrie</b>	<b>53</b>
A.1 Homogene Koordinaten . . . . .	53
A.2 Transformationsmatrizen . . . . .	54
<b>B Verwendete Modelle</b>	<b>57</b>
<b>Literaturverzeichnis</b>	<b>58</b>

# Abbildungsverzeichnis

2.1	Ansichtsbezogenes Verfahren . . . . .	5
2.2	Ablaufschema der objektbezogenen Erkennung . . . . .	7
3.1	Kontur eines Objektes . . . . .	14
3.2	Ablaufschema der Modellaufbereitung . . . . .	15
3.3	Facettendarstellung . . . . .	16
3.4	Tessellierung . . . . .	16
3.5	Unterteilung der Facettenkanten . . . . .	16
3.6	Modellkante . . . . .	17
3.7	konvexe Kanten . . . . .	18
3.8	<i>Kontur</i> -Regel . . . . .	19
3.9	<i>Sichtbarkeitskegel</i> -Regel . . . . .	21
4.1	Arbeitsschritte der Erkennung . . . . .	24
4.2	Kantenextraktion . . . . .	25
4.3	Kanten-Distanz-Bild . . . . .	27
4.4	Projektion . . . . .	29
4.5	Projektionsoptimierung . . . . .	31

5.1	Klassenübersicht . . . . .	39
6.1	Resultat der Modellaufbereitung: Beispiel 1 . . . . .	42
6.2	Resultat der Modellaufbereitung: Beispiel 2 . . . . .	42
6.3	Resultat der Modellaufbereitung: Beispiel 3 . . . . .	42
6.4	Resultat der Modellaufbereitung: Beispiel 4 . . . . .	43
6.5	Verlauf der Projektionsoptimierung . . . . .	44
6.6	Konvergenz der Winkel . . . . .	46
6.7	Projektionsoptimierung . . . . .	46
6.8	Klassifikation des Modells „Fax“ . . . . .	47
6.9	Klassifikation des Modells „Diskettenbox“ . . . . .	48

# Kapitel 1

## Einleitung

Im Bereich des Bildverstehens und der Bildanalyse werden bereits seit vielen Jahren Methoden der modellbasierten Objekterkennung eingesetzt. Ihnen ist gemeinsam, dass in einem Bild oder einer Bildsequenz nach a priori bekannten Objekten gesucht wird. Diese werden klassifiziert und ihre Lage im Raum wird ermittelt. Die Anwendungsfelder der modellbasierten Objekterkennung sind breit gefächert. Es gibt zum Beispiel Systeme, die zur Roboter- oder Fahrzeugnavigation eingesetzt werden. In der industriellen Fertigung kommen modellbasierte Ansätze ebenfalls zum Einsatz und ein weiteres Feld ist die Straßenverkehrsanalyse. Grundsätzlich eignen sich also alle Bereiche, in denen bestimmte, zu erwartende dreidimensionale Objekte durch visuelle Systeme erkannt werden müssen, für den Einsatz eines modellbasierten Ansatzes.

Die Umsetzung dieser Technik weist eine große Vielfalt auf, die schon bei der Modellrepräsentation beginnt und sich über unterschiedliche Ansätze der Merkmalsauswahl bis zu einer großen Auswahl von Klassifikatoren hinzieht. Ein kleiner Überblick über Forschungsansätze in diesen Bereichen wird in Kapitel 2 gegeben. Die vorliegende Arbeit präsentiert ein System zur modellbasierten Objekterkennung auf Basis von dreidimensionalen Modellvorlagen und einem monokularen Bildverarbeitungssystem. Es umfasst sowohl die Aufbereitung der 3D-Modelle, als auch die anschließende Erkennung durch einfache Bild- und Modellmerkmale. Zur experimentellen Untersuchung der Verfahren liegen unter anderem Modelle

typischer Büroeinrichtungsgegenstände vor, welche den Einsatz des Erkennungssystems im Bereich der visuellen Perzeption eines autonom agierenden Roboters demonstrieren. Hierbei kann eine modellbasierte Objekterkennung sowohl für das Auffinden bekannter Objekte, wie auch für die Navigation eingesetzt werden.

Die bereits erwähnte Modellrepräsentation und -aufbereitung ist Thema von Kapitel 3. Hier geht es insbesondere um die automatische Auswahl und Generierung von Modellmerkmalen und den in der vorliegenden Arbeit beschriebenen Ansatz zur Erstellung von Konturmerkmalen aus dreidimensionalen Modellen. Der daran anschließende Abschnitt behandelt die eigentliche Erkennungsaufgabe, also die Klassifikation des im Bild befindlichen Objektes, wie auch die Ermittlung seiner Lage im Raum. In Kapitel 5 werden einige Anmerkungen zur Realisierung des Projektes und zur Software-Architektur angeführt und in Kapitel 6 Resultate der Modellvorbereitung und der Erkennung präsentiert. Abschließend erfolgt eine kurze Zusammenfassung und ein Ausblick, der Erweiterungen und Verbesserungen des Systems thematisiert.

# Kapitel 2

## Erkennung räumlicher Modelle

Viele Methoden und Lösungsansätze im Bereich der modellbasierten Objekterkennung sind schon eingehend untersucht worden. Es wird im Folgenden ein kurzer Überblick über die grundsätzlichen Vorgehensweisen und die damit verbundenen Probleme gegeben, um eine Einordnung dieser Arbeit zu ermöglichen. Hierbei werden nur Arbeiten berücksichtigt, die sich auf die Analyse von Bilddaten stützen, und keine weiteren Sensordaten (wie z.B. von Entfernungsmessern, Sonar, ... ) benötigen. Ein umfassenderer Überblick ist in [Pop94] zu finden.

Allen Methoden ist im Prinzip gemein, dass versucht wird, ein bekanntes dreidimensionales Modell eines Objektes in einem Bild zu detektieren, zu klassifizieren und seine Lageparameter (Raumwinkel und Position) zu ermitteln. Da ein Bild lediglich eine projektive Abbildung eines Objektes darstellt und diese auf Grund der Dimensionsreduktion nicht notwendigerweise informationserhaltend ist, ergibt sich das Problem von Mehrdeutigkeiten in der Interpretation der Abbildung. Alle Verfahren können also für jedes Bild nur eine mehr oder weniger gute Hypothese über das im Bild befindliche Objekt abgeben.

Zur Bestimmung der Ähnlichkeit zwischen dem aktuellen Bild und einem Modell wird ein geeignetes Distanzmaß benötigt. Dieses Maß wird auf Basis von Merkmalen in Modell und Bild errechnet. Auf Merkmalsberechnung und Distanzmaß wird in den Abschnitten 2.5 und 2.6 eingegangen.

Hauptsächlich kann man drei Ansätze der modellbasierten Objekterkennung aus

visueller Information unterscheiden:

- Dreidimensionale Klassifikationsverfahren
- Ansichtsbezogene Verfahren
- Objektbezogene Verfahren

Diese verschiedenen Ansätze werden nun im Folgenden kurz erläutert.

## 2.1 Dreidimensionale Klassifikationsverfahren

Die dreidimensionalen Klassifikationsverfahren versuchen, durch eine entsprechende Vorverarbeitung die im Bild fehlende Tiefeninformation zu gewinnen. Häufig finden dabei weitere Sensoren Verwendung, um zum Beispiel durch Lasertriangulation [HH94] oder Sonar die Tiefeninformation durch andere Wege als die der reinen Bildverarbeitung zu ermitteln. Des Weiteren ist die Verwendung von Stereokamerasystemen zur Tiefenbestimmung möglich.

Mit diesen Verfahren wird schon in der Vorverarbeitung eine dreidimensionale Modellapproximation ermittelt, welche direkt mit dem bekannten Objektmodell verglichen werden kann. Da die Klassifikation im dreidimensionalen Raum erfolgt, haben diese Verfahren auch den Namen *3D/3D-Verfahren*. Sie sollen ebenso wie die nachfolgend beschriebenen ansichtsbezogenen Verfahren nicht näher betrachtet werden, da der zusätzliche Aufwand an Sensorik zu vermeiden ist.

## 2.2 Ansichtsbezogene Verfahren

Die ansichtsbezogenen Verfahren bilden den direkten Gegenpol zu den angesprochenen dreidimensionalen Klassifikationsverfahren. Der Modellvergleich findet hier nicht im Dreidimensionalen, sondern im Bild statt. Zu diesem Zweck wird



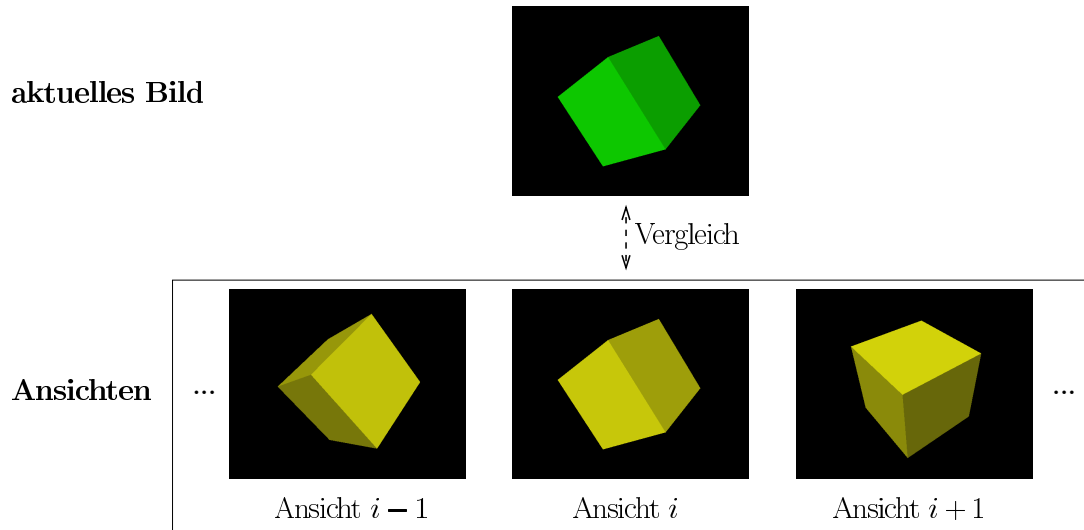


Abbildung 2.1: Ansichtsbezogenes Verfahren

eine Menge diskreter Ansichten vom Modell gespeichert, welche aus der Projektion eines 3D-Modells oder auch aus einer Menge von Aufnahmen des Objektes, ermittelt werden. Diese Ansichten werden mit dem aktuellen Bild verglichen und die bestpassende Ansicht ausgewählt (siehe Abbildung 2.1).

Da sowohl das Modell als auch das Bild zweidimensional repräsentiert sind, wird dieses Vorgehen auch als *2D/2D-Verfahren* bezeichnet. Es ist leicht zu realisieren, da keine dreidimensionalen Modelle der zu erkennenden Objekte gegeben sein müssen, sondern eine Menge verschiedener Ansichten ausreicht. Darin begründet liegen aber auch Schwächen des Verfahrens. Objekte können nur aus den gespeicherten Perspektiven korrekt erkannt werden. Außerdem ist die Anwendung des Verfahrens aufwändig, da für jedes bekannte Modell im Prinzip alle Ansichten mit dem aktuellen Bild verglichen werden müssen. Durch Auswertung von Objektsymmetrien und Rotationsinvarianzen kann die Anzahl der benötigten Ansichten allerdings reduziert werden.

## 2.3 Objektbezogene Verfahren

Beim Ansatz der objektbezogenen Verfahren liegt das Modell dreidimensional vor und wird mittels Projektion mit dem zweidimensionalen Bild verglichen. Aus diesem Grunde spricht man auch von *3D/2D-Verfahren*.

Bezogen auf die ansichtsbezogenen Verfahren aus Abschnitt 2.2 wird hier also nicht in der Vorverarbeitung bereits eine diskrete Anzahl verschiedener Objektperspektiven erzeugt bzw. aufgenommen. Stattdessen wird die jeweils benötigte Ansicht durch Projektion des dreidimensionalen Modells erstellt. Im Gegensatz zum 3D/3D-Verfahren (Abschnitt 2.1) ist jedoch keine aufwändige Sensorik notwendig, da direkt mit dem zweidimensionalen Bilddaten einer Kamera eine Erkennung realisiert werden kann. Das Vorgehen bei der objektbezogenen Erkennung lässt sich grob in vier wesentliche Teile gliedern, wie es auch in Abbildung 2.2 wiedergegeben ist.

Wesentlicher Unterschied zu dem 2D/2D-Verfahren ist die Notwendigkeit einer initialen Lagehypothese. Weil das Verfahren auf die Projektion des Modells aus einer bestimmten Perspektive angewiesen ist, muss diese Perspektive – zumindest grob – bekannt sein. Auf dieses Problem wird in Abschnitt 2.4 noch genauer eingegangen. Da diese Hypothese jedoch nicht exakt sein kann, ist es erforderlich eine Optimierung der Projektion vorzunehmen, um den Vergleich zwischen Bild und Modell bei korrekter Abbildung vornehmen zu können.

Dieser Optimierungsprozess besteht aus der Projektion des Modells auf Basis der aktuellen Lagehypothese und der Bestimmung einer neuen optimierten Projektion. Bei der Projektion werden nur die Merkmale des Modells abgebildet, die auf Grund der aktuellen Hypothese im Bild zu erwarten sind.

Eine solche optimierte Projektion kann für alle bekannten Modelle errechnet werden, so dass im Anschluss eine Klassifikation der im Bild befindlichen Objekte ohne die Ergebnisverfälschung einer inkorrekten Projektion möglich wird.

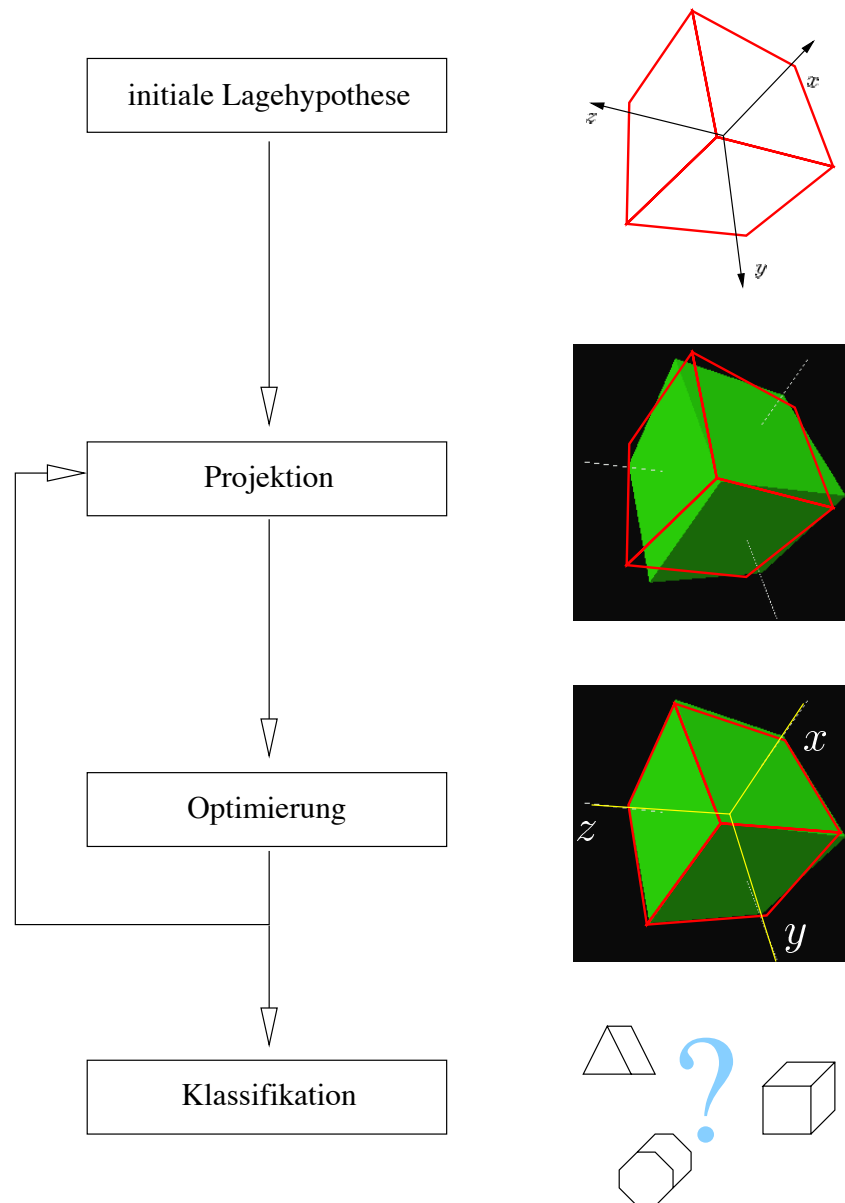


Abbildung 2.2: Ablaufschema der objektbezogenen Erkennung

## 2.4 Initiale Lagehypothese

Die initiale Lagehypothese spielt bei der Erkennung auf Basis des objektbezogenen Ansatzes eine sehr entscheidende Rolle, da auf ihrer Grundlage die erste Projektion des Modells erfolgt. Auch wenn im folgenden Optimierungsprozess Fehler der Initialisierung ausgeglichen werden können, so ist doch zu erwarten, dass eine schlechte Hypothese nicht zum erwünschten Erkennungsergebnis führt, die Optimierung also fehlschlägt.

Um zu einer akzeptablen Hypothese zu gelangen, wird meist Vorwissen genutzt. Dies kann z.B. die Kenntnis über die übliche Lage eines Objektes sein. So ist allgemein anzunehmen, dass sich ein Tisch mit den Füßen auf dem Boden befindet, so dass also nur noch einer statt drei Rotationsfreiheitsgraden zu beachten ist. Ebenso kann bei bewegten Objekten durch Analyse von Bildsequenzen diese Bewegung ermittelt werden (vgl. [Kol93] für ein Beispiel sich bewegender Autos). Aus der Kenntnis der Richtung und dem Vorwissen einer typischen Ausrichtung des Objektes während dieser Bewegung (Fahrtrichtung eines Autos) kann eine gute Lagehypothese gewonnen werden.

Im Falle des Einsatzes der modellbasierten Objekterkennung zur Orientierung eines autonomen mobilen Systems (AMS) in einer bekannten Umgebung, wie zum Beispiel in [LZ96] vorgestellt, kann die initiale Hypothese auch auf Basis der letzten Orientierung oder der Kenntnis einer Karte der Umgebung erfolgen.

Ist eine Hypothesengenerierung auf Basis von Vorwissen oder aus Bewegungsinformation nicht möglich, so ist die Initialisierung schwer zu realisieren. Hier kann man sich oft nur auf den ansichtsbezogenen Ansatz aus Abschnitt 2.2 zurückziehen, also eine diskrete, endliche Menge von Projektionen des Modells mit dem Bild vergleichen.

## 2.5 Merkmale

Merkmale dienen dazu, die für eine Objekterkennung relevanten Charakteristika von Modell und Bild hervorzuheben und die Repräsentation des Objektes soweit

zu vereinfachen, dass eine effiziente Erkennung möglich wird. Im Bereich der modellbasierten Objekterkennung auf Basis visueller Informationen haben sich vor allem Kantenmerkmale zur Erkennung durchgesetzt, welche in vielen Arbeiten ([Kol93], [LZ96], ... ) zum Einsatz kommen und auch in der vorliegenden genutzt werden. Sie haben den Vorteil, aus Intensitätsbildern leicht berechenbar, und auch im Modell einfach repräsentierbar zu sein. Ihre Charakteristika sind in wenigen Parametern erfassbar, so dass die Berechnung von Distanzen und die Optimierung effizient realisierbar sind. Leider sind Kanten keine besonders robusten Merkmale, da in einer realen Umgebung durch Rauschen, Schattenwurf und andere Objekte im Bild viele nicht durch das Modell erklärbare Kanten auftreten oder erwartete Kanten nicht sichtbar sind.

Im Bereich der Bildverarbeitung sind noch etliche weitere Merkmale bekannt, welche die Charakteristika von Objekten wiedergeben, zum Teil aber weitaus komplexer sind. Hierzu gehören zum Beispiel Texturmerkmale, wie Grauwertübergangsmatrizen, oder auch Farbmerkmale.

## 2.6 Distanzmaße

Von herausragender Bedeutung bei der Objekterkennung ist die Bestimmung eines geeigneten Distanzmaßes mit welchem sich die Ähnlichkeit zwischen Modell und Bild des Objektes erfassen lässt. Folgende Anforderungen werden im Allgemeinen an ein geeignetes Distanzmaß gestellt:

- **Monotonie:** Die Distanz muss bei zunehmender Ähnlichkeit kleiner werden.
- **Positivität:** Die Distanz ist immer positiv, eine Distanz gleich 0 entspricht der Identität.
- **einfache Berechenbarkeit:** Die Distanz sollte effizient errechenbar sein.

In der Regel wird die Distanz nicht direkt auf den Bild- bzw. Modelldaten errechnet, sondern aus zuvor ermittelten Merkmalen. Wenn solche Merkmale als

Vektoren  $\vec{a}$  und  $\vec{b}$  gegeben sind, dann kann man auf ihnen Distanzmaße wie die euklidische Norm

$$d_E(\vec{a}, \vec{b}) = \sqrt{\sum_{i=0}^N (a_i - b_i)^2} \quad (2.1)$$

oder die Mahalanobis-Distanz

$$d_M(\vec{a}, \vec{b}) = (\vec{a} - \vec{b})^T \Sigma^{-1} (\vec{a} - \vec{b}), \quad (2.2)$$

welche durch die Kovarianzen  $\Sigma$  der Merkmalskomponenten normiert ist, definieren.

Zusätzlich zu diesen allgemeinen, in der Mustererkennung etablierten, Distanzmaßen gibt es weitere, die spezieller auf die Art der verwendeten Merkmale ausgerichtet sind. So gibt es spezielle Kanten-, Struktur- und Farbdistanzmaße, denen keiner dieser allgemeinen mathematischen Ansätze zu Grunde liegt. In Abschnitt 4.2 wird mit dem Kanten-Distanz-Bild ein solches Verfahren für Konturkanten vorgestellt.

## 2.7 Projektionsoptimierung

Die Projektionsoptimierung ist vor allem abhängig von der Wahl des Distanzmaßes, da auf seiner Basis die jeweils korrespondierenden Modell- und Bildmerkmale ermittelt und ihr Abstand voneinander bestimmt wird. Meist wird als Korrespondenzkriterium der minimale Abstand zwischen den beiden Merkmalen verwendet und dann eine Projektion errechnet, welche die Summe der Distanzen zwischen den korrespondierenden Merkmalen zu minimieren versucht. Zu unterscheiden ist die globale Optimierung der Projektion, welche dazu dient eine initiale Hypothese (Abschnitt 2.4) zu optimieren, und die lokale Optimierung, welche die lokale Distanz jeder einzelnen Korrespondenz minimiert. Letztere kann dazu dienen, verformbare Modelle, wie sie zum Beispiel D. Koller in seiner Arbeit [Kol93] benutzt, zu ermitteln und auf Basis der Verformungsparameter eine Klassifikation zu realisieren. Die globale Optimierung ist bei nahezu allen Verfahren notwendig, da sie zur Ermittlung der Lage des Objektes dient.

Zur Realisierung der Optimierung gibt es verschiedene Ansätze: Einige verfolgen die Möglichkeit, die Parameter der Projektion vom Dreidimensionalen ins Zweidimensionale inklusive der Kameraparameter zu optimieren. Häufig wird nur eine Optimierung innerhalb des Bildes – also im zweidimensionalen Raum – angestrebt, wobei eine Rekonstruktion der genauen dreidimensionalen Lage des Objektes nicht mehr möglich ist. Die Klassifikation des Objektes auf Basis der Verformungsparameter ist jedoch weiterhin gut realisierbar. Ist die Lage im Raum als Ergebnis von Interesse, so muss eine Optimierung auf Basis der Projektion erfolgen, wie es zum Beispiel in [LZ96] geschieht.

## 2.8 Modellklassifikation

Die Entscheidung, welches Modell im Bild zu sehen ist, wird durch eine Klassifikation realisiert. Durch die Definition eines Abstandsmaßes ist die einfachste Klassifikationsmethode direkt gegeben: Die bekannten Modelle sind Prototypen für die Verwendung eines *Nächsten-Nachbarn-Klassifikators*, welcher die Entscheidung direkt bezüglich des definierten Distanzmaßes aus Abschnitt 2.6 trifft.

Des Weiteren ist eine Klassifikation auf Basis der Verformungsparameter, die sich aus der Optimierung des Modells bezüglich des aktuellen Bildes ergeben, oder den Merkmalen möglich. Hier können klassische numerische Klassifikatoren wie Polynomklassifikatoren oder neuronale Netze [Zel97] zum Einsatz kommen.

## 2.9 Ziele dieser Arbeit

Die Ansätze zur Realisierung einer modellbasierten Objekterkennung sind sehr vielseitig. Es soll in dieser Arbeit ein einfacher Ansatz vorgestellt und realisiert werden, der ausgehend von einer 3D/2D-Erkennung die Möglichkeit bietet, mit einfachen Merkmalen, Distanzmaßen und Optimierungsalgorithmen eine Erkennungsaufgabe zu lösen. Das Problem der initialen Lagehypothese wird hier nicht eingehender erörtert. Es sollen aus gegebenen Modelldaten Modell-Merkmale automatisch erzeugt und auf ihrer Basis Intensitätsbilder klassifiziert

werden können. Dabei soll aus den zweidimensionalen Aufnahmen auch die Position und Lage des Objektes im Bild ermittelt werden. Die Architektur des Systems ist so anzulegen, dass eine leichte Erweiterung auf andere Merkmale und Maße möglich ist und eine weitgehende Unabhängigkeit vom Datenformat der dreidimensionalen Modelle vorliegt.



# Kapitel 3

## Modellaufbereitung

Bevor Objekte in einem Bild detektiert und klassifiziert werden können, müssen diese in einer dem Computer effizient zugänglichen Form repräsentiert sein. Da bei der Erkennung meist hohe Anforderungen an die Geschwindigkeit der Algorithmen gestellt werden, sollte die Vorverarbeitung der Objekt-Modelle bereits Berechnungen (offline) durchführen, deren Ergebnisse das Erkennungssystem (online) nutzt.

Außerdem sollen nicht die gesamten Modellinformationen zur Erkennung herangezogen, sondern stattdessen die relevanten und für das entsprechende Modell signifikanten Merkmale ausgewählt werden, welche im Folgenden als *Identifikationsmerkmale* bezeichnet werden. Es handelt sich dabei um lokalisierte Merkmale, die die geometrische Eigenschaft eines Teils des Modells beschreiben und in das Bild projizierbar sind.

Die Auswahl dieser Merkmale kann manuell unter Nutzung von Expertenwissen erstellt werden. Dabei ist darauf zu achten, dass das Modell unter der Menge aller Projektionen gut repräsentiert wird – also aus jedem Blickwinkel eine Erkennung auf der Basis der sichtbaren Identifikationsmerkmale möglich ist. Diese Auswahl durch einen Experten vornehmen zu lassen ist jedoch sehr umständlich, und es ist wünschenswert den Vorgang dieser Auswahl zu automatisieren, wie es in dieser Arbeit realisiert wird.

Darüberhinaus besteht das Problem, dass die Modelldaten aus unterschiedlichsten

Quellen kommen können. Dreidimensionale Modelldaten können aus 3D-Scanner, CAD-Anwendungen oder schon vorgefertigten Modellen aus anderen digitalen Quellen (z.B. Internet) gewonnen werden. Das bedeutet, dass eine Umwandlung der unterschiedlichsten Datenformate in ein für den Erkenner geeignetes notwendig ist, wobei nur die für die Klassifikation benötigten Informationen selektiert werden sollen.

Dieses Kapitel stellt die Lösungsansätze einer einfachen Modellaufbereitung zur modellbasierten Erkennung auf Basis der Kontur der Objekte vor. Als Kontur wird hier der *Schattenriss* eines Objektes bezeichnet, wie in Abbildung 3.1 zu sehen ist. Im Folgenden wird der Ablauf der Modellaufbereitung skizziert. Anschließend wird in 3.2 dargelegt, welche Anforderungen an die Daten der Modelle gestellt werden und wie diese vor der eigentlichen

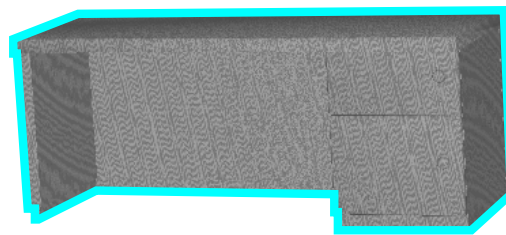


Abbildung 3.1: Die Kontur eines Schreibtisches

Merkmalerstellung, deren Vorstellung in Abschnitt 3.3 erfolgt, aufbereitet werden. Abschnitt 3.4 präsentiert die Realisierung von Berechnungen, die bereits vor der eigentlichen Erkennung durchgeführt werden, am Beispiel von Sichtbarkeitsregeln.

### 3.1 Ablauf der Modellaufbereitung

Der Prozess der Modellaufbereitung gliedert sich in drei wesentliche Abschnitte, die in Abbildung 3.2 zu sehen sind:

1. Formatkonvertierung
2. Generierung von Identifikationsmerkmalen
3. Regelerstellung

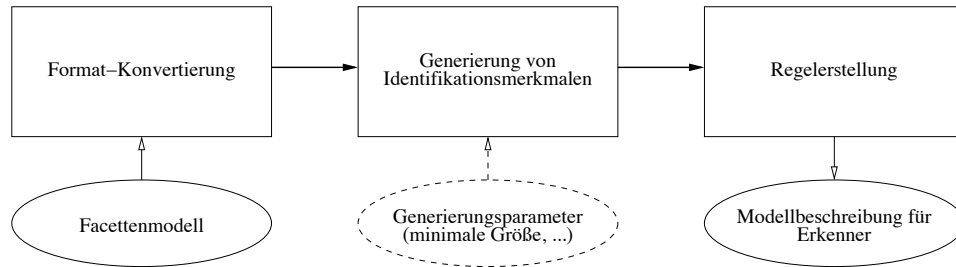


Abbildung 3.2: Ablaufschema der Modellaufbereitung

Der erste Schritt wird durch Verwendung von 3D-Konvertierungsprogrammen (wie z.B. [Bai01]) gelöst und soll hier nicht weiter diskutiert werden. Die Schritte zwei und drei sind Teil dieser Arbeit und werden im Folgenden näher erläutert, wobei ihre Bedeutung für die Erkennung dargestellt wird. Die in der Abbildung sichtbaren Generierungsparameter dienen dazu, Einfluss auf die Auswahl der Merkmale zu nehmen, um also beispielsweise aus Erfahrung ungeeignete Merkmale auszuschließen.

## 3.2 Modelldaten

In dem hier vorgestellten System geht die Modellaufbereitung von dreidimensionalen Facettenmodellen aus, also Modellen, die nur durch die Angabe von Flächen spezifiziert sind. Diese Einschränkung zieht nach sich, dass Objekte mit runden Formen durch eine Menge von ebenen Flächen approximiert werden müssen. Diese Vereinfachung des Modells kann für die Erkennung auf Basis der Kontur durchaus legitimiert werden.

Die einzelnen Facetten sind durch Angabe von Punkten im 3D-Raum (*vertices*) und den sie verbindenden Kanten repräsentiert (Abbildung 3.3). Zur einfacheren Handhabung werden die Facetten in einem ersten Vorverarbeitungsschritt tesselliert, d.h. in mehrere Dreiecke zerlegt, wobei die Drehrichtung (also die Nummerierung der Punkte) der Flächen erhalten bleibt (vergleiche Darstellung 3.4). Diese Drehrichtung ist von Bedeutung, da durch sie die Richtung der Flächen-

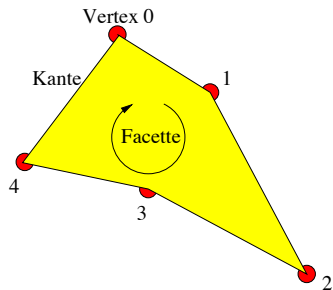


Abbildung 3.3: Darstellung einer Facette durch die Angabe ihrer Eckpunkte:  $\{0, 1, 2, 3, 4\}$

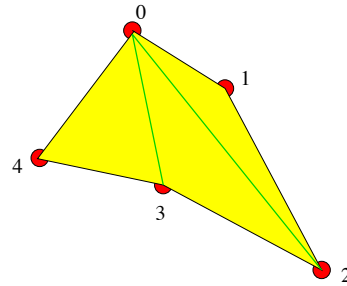


Abbildung 3.4: Tessellierung der Facette durch drei Dreiecke:  $\{\{0, 1, 2\}, \{0, 2, 3\}, \{0, 3, 4\}\}$

normalen definiert ist, welche die Außenseite des Objektes festlegt: Die Normale einer Facette zeigt immer nach außen.

Da in dem hier präsentierten Ansatz Kanten des Facettenmodells als potenzielle Identifikationsmerkmale verwendet werden sollen, wird aus dem gegebenen, tesselierten Facettenmodell ein Gittermodell errechnet. Dabei werden Kanten, die durch mehrere Punkte der Facetten verlaufen, unterteilt, so dass anschließend jede Kante nur durch einen Start- und Endpunkt definiert ist und keine weiteren Punkte mehr überstreicht. Abbildung 3.5 zeigt dieses Vorgehen anhand von fünf dreieckigen Facetten, die vier einzelne Kanten erzeugen. Auf diese Weise sind pro Kante genau zwei Flächen gegeben, die diese Kante bilden, was für die weitere Verarbeitung von Bedeutung ist.

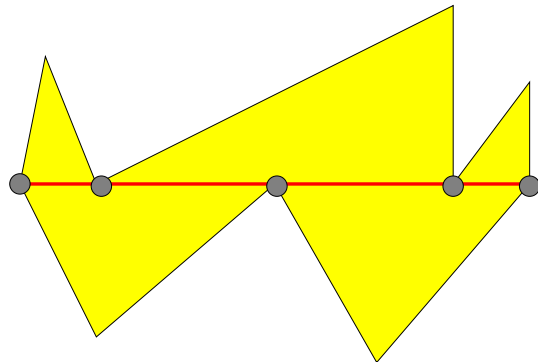


Abbildung 3.5: Unterteilung in vier Kanten mit exakt zwei Flächen

### 3.3 Auswahl von Identifikationsmerkmalen

Wie bereits in der Einleitung dieses Kapitels angesprochen, sollen nicht die gesamten Modellinformationen zur Erkennung herangezogen werden. Stattdessen erfolgt eine automatische Auswahl von Identifikationsmerkmalen, auf denen die Erkennung basiert. Wie diese Selektion realisiert werden kann, wird im Folgenden erläutert.

Da in der vorliegenden Arbeit eine Objekterkennung auf Basis der Konturinformation der Projektion von dreidimensionalen Objekten durchgeführt wird, resultiert daraus, dass in der Modellaufbereitung konvexe Kanten des Modells als Identifikationsmerkmale gesucht werden müssen. Man kann sich leicht vor Augen führen, dass konkave Kanten eines Objektes nicht zur Kontur beitragen. Die Auswahl konvexer Kanten auf der Grundlage eines Facettenmodells ist durch Analyse der Winkel der Normalen der beiden an eine Kante angrenzenden Flächen möglich. In Abbildung 3.6 ist eine Kante zu sehen, die durch zwei Flächen mit den entsprechenden Normalen ( $\vec{n}_1$  und  $\vec{n}_2$ ) gebildet wird. Die Ermittlung konvexer Kanten im Modell geschieht durch die Addition einer kleinen Distanz  $\epsilon$  in Richtung sowie in Gegenrichtung der beiden Normalen  $\vec{n}_1$  und  $\vec{n}_2$  auf den jeweiligen Mittelpunkt  $m_1$  und  $m_2$  beider Flächen (Abbildung 3.7):

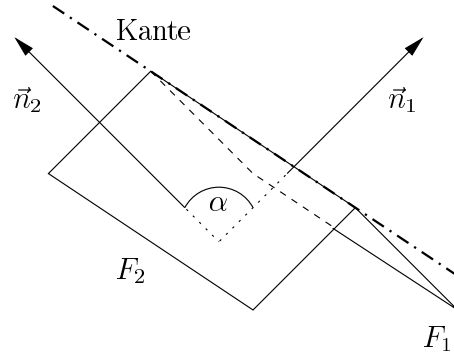


Abbildung 3.6: Darstellung einer Modellkante mit den angrenzenden Flächen

$$\begin{aligned}\vec{a} &= (\vec{m}_1 + \vec{n}_1\epsilon) - (\vec{m}_2 + \vec{n}_2\epsilon) \\ \vec{b} &= (\vec{m}_1 - \vec{n}_1\epsilon) - (\vec{m}_2 - \vec{n}_2\epsilon)\end{aligned}\tag{3.1}$$

Ist die Länge des Vektors  $\vec{a}$  größer als die von  $\vec{b}$ , so ist die Kante konvex, da die Normale einer Kante nach Voraussetzung immer nach außen zeigt.

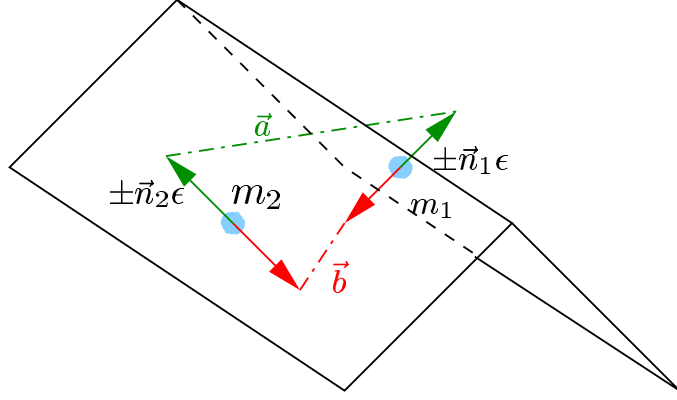


Abbildung 3.7: Ermittlung konvexer Kanten II

Darüberhinaus ist die Berechnung der Krümmung der Kante durch die Normalen möglich. Der Winkel  $\alpha$ , den die beiden Normalen  $\vec{n}_1$  und  $\vec{n}_2$  einschließen kann durch

$$\alpha = \arccos \frac{\vec{n}_1 \cdot \vec{n}_2}{|\vec{n}_1| |\vec{n}_2|} \quad (3.2)$$

errechnet werden. Kanten mit sehr kleinem  $\alpha$  sind als Kontur nur sehr eingeschränkt sichtbar. Daher kann ein Wert  $\alpha_{min}$  als Generierungsparameter angegeben werden, so dass Identifikationsmerkmale nur erzeugt werden, wenn auch

$$\alpha_{min} < \alpha \quad (3.3)$$

gilt. Auf diese Weise ist die Auswahl von Kanten zur Erkennung auf Basis von Konturinformation möglich.

### 3.4 Sichtbarkeitsregeln

Auf Grund der Projektion vom dreidimensionalen in den zweidimensionalen Raum ergibt sich die Notwendigkeit der Ermittlung verdeckter Identifikationsmerkmale. Dies kann zum Teil bereits in der Aufbereitung der Modelle durch

Festlegung von *Sichtbarkeitsregeln* geschehen. Diese bieten durch Angabe geometrischer Parameter die Möglichkeit, aus der Projektionshypothese die Merkmale zu entfernen, die begründet durch physikalische Gegebenheiten wie Verdeckung, Schattenwurf und zu große Entfernung nicht im Bild erwartet werden können.

Im Folgenden werden zwei einfache Sichtbarkeitsregeln für Konturkanten vorgestellt. Beide modellieren Verdeckungen, die durch die Modellgeometrie gegeben sind. Probleme durch Verdeckungen anderer im Bild befindlicher Objekte sind so allerdings nicht zu lösen.

### 3.4.1 Kontur-Regel

Die *Kontur-Regel* stellt die einfachste der Sichtbarkeitsregeln dar. Sie berechnet die Verdeckung einer Kante durch eine der beiden angrenzenden Flächen. Dies geschieht auf Basis eines Sichtstrahls  $\vec{s}$  und der beiden Flächennormalen  $\vec{n}_1$  und  $\vec{n}_2$ . Wenn eine der beiden Normalen in Richtung und die andere in die Gegenrichtung des Sichtstrahls zeigt, so ist die Kante als Kontur im Bild sichtbar (vergleiche Abbildung 3.1), andernfalls nicht. Mathematisch stellt sich diese Aussage wie folgt dar:

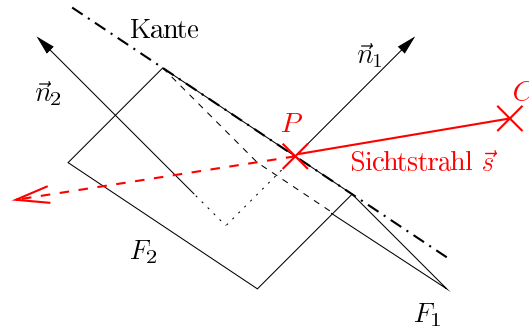


Abbildung 3.8: Anwendung der *Kontur-Regel*

$$\underbrace{\text{sgn}(\vec{n}_1 \cdot \vec{s})}_{\text{Richtung von } \vec{n}_1 \text{ zu } \vec{s}} \neq \underbrace{\text{sgn}(\vec{n}_2 \cdot \vec{s})}_{\text{Richtung von } \vec{n}_2 \text{ zu } \vec{s}} \quad (3.4)$$

Während der Erkennung werden zur Auswertung der Regel die beiden Normalen sowie der besagte Sichtstrahl benötigt. Die Normalen können bereits in der hier diskutierten Modellaufbereitung errechnet werden, so dass der Erkennungsalgorithmus nur noch den Sichtstrahl  $\vec{s}$  berechnen und die Ungleichung 3.4 prüfen

muss. Die Berechnung von  $\vec{s}$  ergibt sich aus dem Mittelpunkt  $P$  der Kante und der Position  $C$  der Kamera.

$$\vec{s} = \overrightarrow{CP} \quad (3.5)$$

Eine Verdeckung durch nicht angrenzende Flächen kann durch die *Kontur*-Regel nicht modelliert werden. Die Menge noch zu überprüfender Merkmale kann durch sie jedoch deutlich reduziert werden, so dass nur noch für eine geringere Anzahl aufwändigere Regeln überprüft werden müssen, was einen deutlichen Effizienzgewinn bringt. Eine Regel, die in der Lage ist alle Verdeckungen eines einzelnen Objektes zu modellieren, ist die im Anschluss vorgestellte *Sichtbarkeitskegel*-Regel.

### 3.4.2 Sichtbarkeitskegel-Regel

Diese Regel gibt direkt Sichtbarkeitsbereiche für das Identifikationsmerkmal an. Zu diesem Zweck erzeugt die Modellaufbereitung einen oder mehrere Kegel  $K_i$  mit einer jeweiligen Hauptachse  $\vec{c}_i$  und einem Winkel  $\alpha_i$ . Liegt der, in Gleichung 3.5 definierte, Sichtstrahl  $\vec{s}$  innerhalb des Kegels, wenn also

$$\arccos \frac{\vec{c}_i \cdot \vec{s}}{|\vec{c}_i| |\vec{s}|} < \alpha_i \quad (3.6)$$

gilt, dann ist das Merkmal sichtbar. Dieser Fall ist in Abbildung 3.9 auf der rechten Seite zu sehen. Links hingegen liegt der Sichtstrahl außerhalb des Kegels, das Merkmal ist also aus der gegebenen Perspektive nicht sichtbar.

Während der Modellaufbereitung ist die Berechnung solcher Sichtbarkeitsbereiche sehr aufwändig. Sie wird durch einen Algorithmus realisiert, der das Modell aus einer diskreten Menge vieler verschiedener Perspektiven analysiert. Aus jeder Perspektive wird für den jeweiligen Sichtstrahl der Schnitt mit allen Facetten des Modells errechnet (Schnitt von Gerade und Ebene siehe beispielsweise in [Jen94]). Existiert ein Schnittpunkt mit einer der Facetten, so ist dieses Merkmal bei dieser Ansicht nicht sichtbar. Auf diese Weise ergibt sich für jedes Merkmal eine Menge von Sichtbarkeitsbereichen, die, damit sie einfach zu parametrisieren sind, durch eine Menge von Kegeln approximiert werden. Dadurch wird die Sichtbarkeit zwar



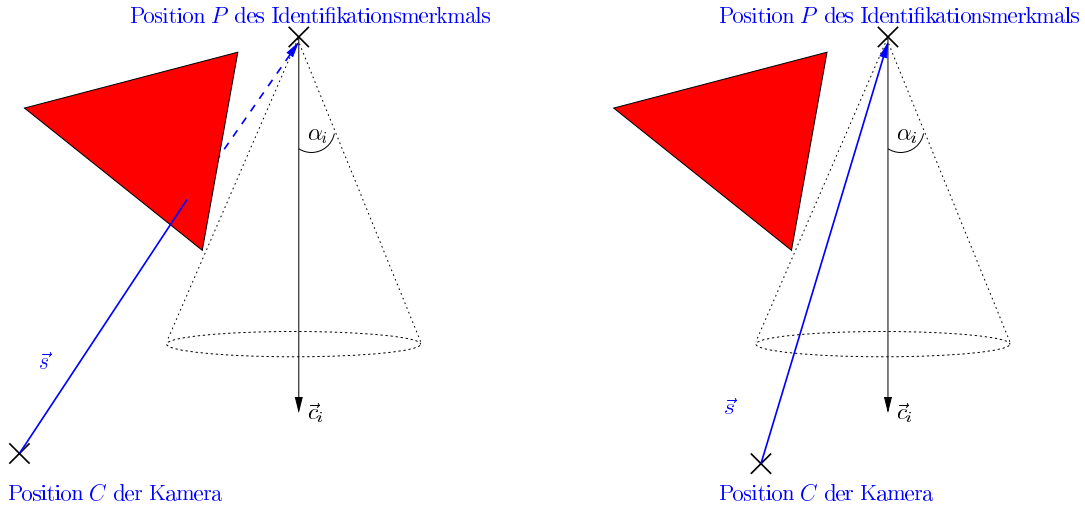


Abbildung 3.9: Prinzip der *Sichtbarkeitskegel*-Regel – Im rechten Bild ist das Merkmal sichtbar, im linken wird es durch eine Facette verdeckt

mehr eingeschränkt als notwendig, aber in der Praxis hat sich diese Parametrisierung, insbesondere wenn relativ einfache und vorwiegend konvexe Objekte betrachtet werden, bewährt.

Durch das präsentierte Verfahren ist eine Modellaufbereitung möglich, die ausgehend von einem Facettenmodell eines dreidimensionalen Objektes eine automatische Generierung von konvexen Kanten als Identifikationsmerkmale realisiert. Durch Berechnung von Parametern zur Bestimmung der Sichtbarkeiten kann der Aufwand der zeitkritischen Objekterkennung reduziert werden. Aus den Modell-daten werden somit nur die Informationen extrahiert, die für die Erkennung von Bedeutung sind.



# Kapitel 4

## Objekterkennung

Den groben Ablauf eines Algorithmus objektbezogener Erkennungssysteme gab schon Abbildung 2.2 in der allgemeinen Einführung zum Thema 3D/2D-Erkennung wieder. Hier soll auf die realisierten Kontur-Merkmale, die Projektionsoptimierung durch multidimensionale lineare Regression und die Benutzung eines Kanten-Distanz-Bildes als Abstandsmaß eingegangen werden.

Ziel ist es, das Modell mit dem Bild auf Basis der Identifikationsmerkmale zu vergleichen. Dazu müssen sowohl mit dem Modell als auch dem Bild einige Arbeitsschritte durchgeführt werden, welche in Abbildung 4.1 zu sehen sind. Links sieht man die Schritte der Modellverarbeitung – im Einzelnen die initiale Lagehypothese und die Anwendung der in Kapitel 3 vorgestellten Regeln. Auf der rechten Seite dieser Abbildung sind alle Operationen, die mit dem aufgenommenen zweidimensionalen Bild durchgeführt werden, aufgezeigt.

### 4.1 Bildmerkmale

Als Bildmerkmale werden im hier vorgestellten Ansatz Kanten verwendet, also Bereiche mit ausgeprägtem Gradienten im Intensitätsbild. Die Detektion der Kanten wird durch Anwendung eines *Canny*-Operators [Can86] gelöst. Dieser führt im Prinzip drei Schritte durch, um Kanten zu extrahieren:

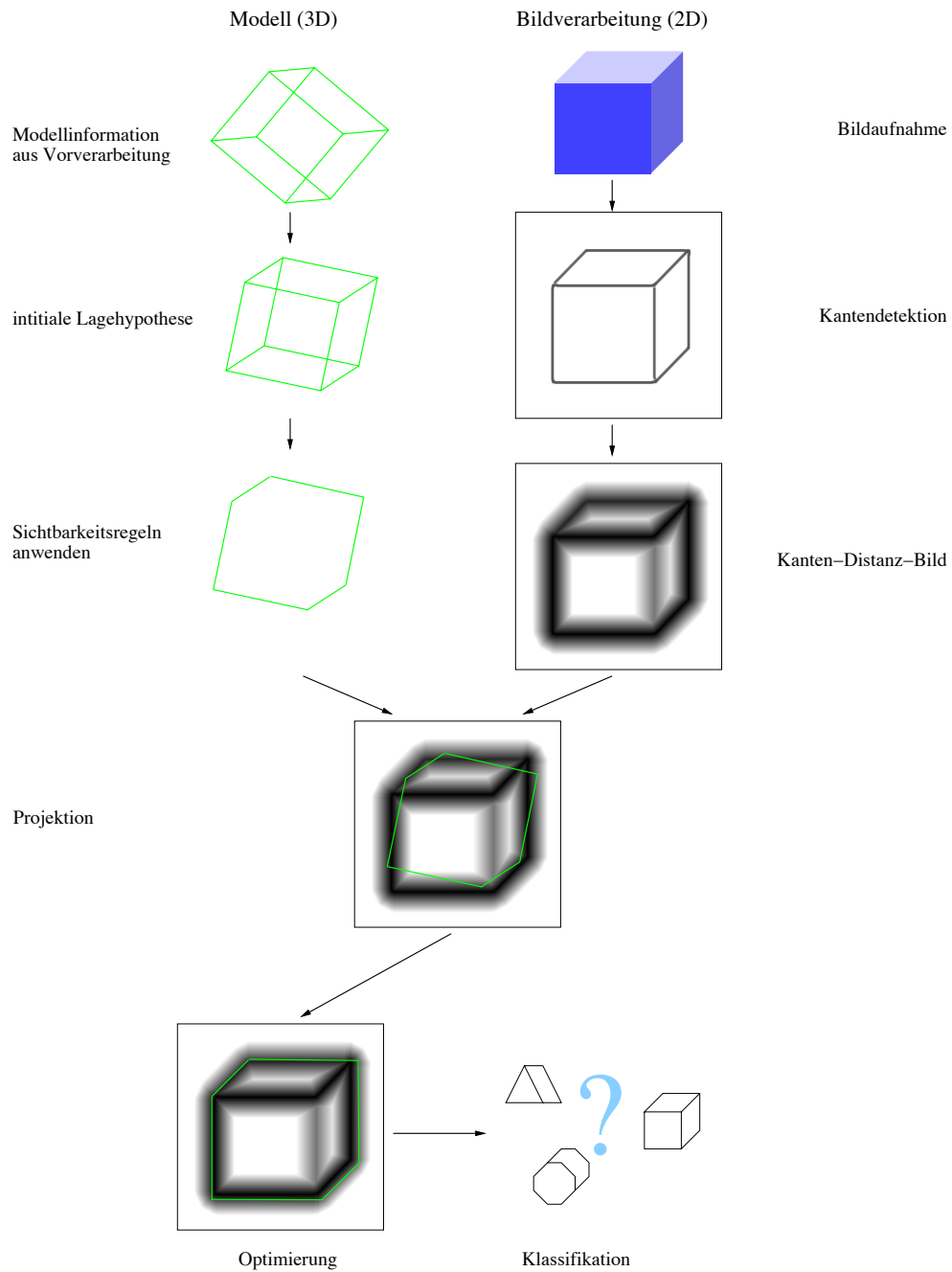


Abbildung 4.1: Arbeitsschritte der Erkennung

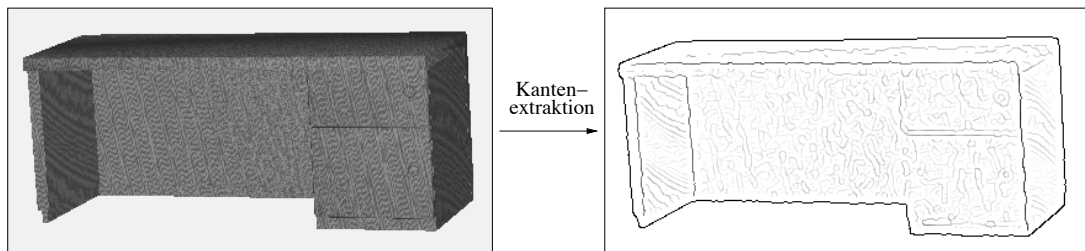


Abbildung 4.2: Anwendung der Kantenextraktion auf dem künstlichen Bild eines Schreibtisches mit Holzmaserung. Das Kantenbild (rechts) ist zur besseren Visualisierung invertiert dargestellt.

1. **Bildglättung:** Durch Gaußmasken wird das Bild geglättet und Rauschen abgeschwächt.
2. **Gradientenbild:** Anwendung eines Sobelfilters auf dem geglätteten Bild zur Erzeugung je eines Gradientenintensitäts- und -richtungsbildes.
3. **Non-Maximum-Suppression:** Elimination von Rauschen durch Anwendung eines Hysterese-Schwellwert-Verfahrens und Ausdünnung der Kanten auf die Stärke eines Pixels.

Das Resultat einer solchen Kantenextraktion ist in Abbildung 4.2 zu sehen.

## 4.2 Kanten-Distanz-Bild

Die Definition eines Abstandsmaßes spielt für die Erkennung eine entscheidende Rolle. Auf Basis des Kantenbildes ist es leicht möglich, parametrisierte Approximationen der Kanten im Bild, beispielsweise durch Geradenstücke, zu errechnen, wie es u.a. auch in der Arbeit von Koller [Kol93] beschrieben ist. Parametrisierte Kanten erfordern jedoch die Benutzung von Schwellwerten zur Entscheidung über die Existenz einer Kante. Liegt auf Grund von z.B. ungünstiger Beleuchtung ein Gradientenwert unter dem gegebenen Schwellwert, so wird die Kante entfernt und ist in den weiteren Verarbeitungsschritten nicht mehr vorhanden. Des

Weiteren erfordert die Projektionsoptimierung die Zuordnung einer Modellkante zur parametrisierten Bildkante. Dies ist insbesondere deshalb problematisch, weil Bildkanten durch Rauschen, Schattenwurf oder andere Bildstörungen in mehrere einzelne parametrisierte Kanten zerfallen können. Ist eine Zuordnung gefunden, so ist auch die Definition eines Distanzmaßes noch relativ schwierig. Am Beispiel der Approximation durch Geradenstücke wird dies deutlich. Kanten sind durch Angabe von Ortsvektor und Orientierungswinkeln gegeben. Der Vergleich zweier Kanten gestaltet sich schwierig, da die Winkelangaben im Bereich zwischen  $-\pi$  und  $\pi$ , die Ortsvektorkoordinaten aber im Prinzip Werte aus beliebigen Intervallen annehmen können. Die Anwendung eines einfachen Distanzmaßes wie zum Beispiel in Gleichung 2.1 definiert, kommt somit nicht in Frage. Hier kann, wie von Koller vorgeschlagen, die Mahalanobis-Distanz aus Gleichung 2.2 verwendet werden.

**Definition** Auf Grund der angesprochenen Probleme wird in dieser Arbeit kein Ansatz parametrisierter Kanten eingesetzt. Stattdessen wird ein pixelbasiertes Distanzmaß definiert, welches für jeden Punkt einer Projektion die Distanz zum Bildkantenpunkt errechnet. Um keine Schwellwerte etablieren zu müssen, aber dennoch Bildkanten, die durch Rauschen entstanden sind, nicht zu stark zu berücksichtigen, wird zur Berechnung der Distanz  $d_K(x, y)$  die Entfernung  $\delta_K(x, y)$  zu einem Kantenpixel  $K = (x', y')$  mit der Kantenausprägung  $I_K$  gewichtet. Diese ist als lokaler Gradient der Intensität durch die Kantenextraktion nach Abschnitt 4.1 gegeben. Auf diese Weise ergibt sich die Distanz eines Punktes  $(x, y)$  zu einem Kantenpixel durch

$$d_K(x, y) = I_{max} - \frac{I_K (\delta_{max} - \delta_K(x, y))}{\delta_{max}}, \quad (4.1)$$

wobei die Berechnung nur bis zu einem maximalen Abstand von  $\delta_{max}$  möglich ist. Der Parameter  $I_{max}$  gibt den maximalen Wert der Kantenausprägung (üblicherweise 255) an.

**Algorithmus** Die Umsetzung in einen Algorithmus erfordert die Analyse des Resultats der Kantendetektion aus Abschnitt 4.1. Das Gradientenbild der Kantendetektion wird zeilen- und spaltenweise abgetastet. Ist an einer Position die

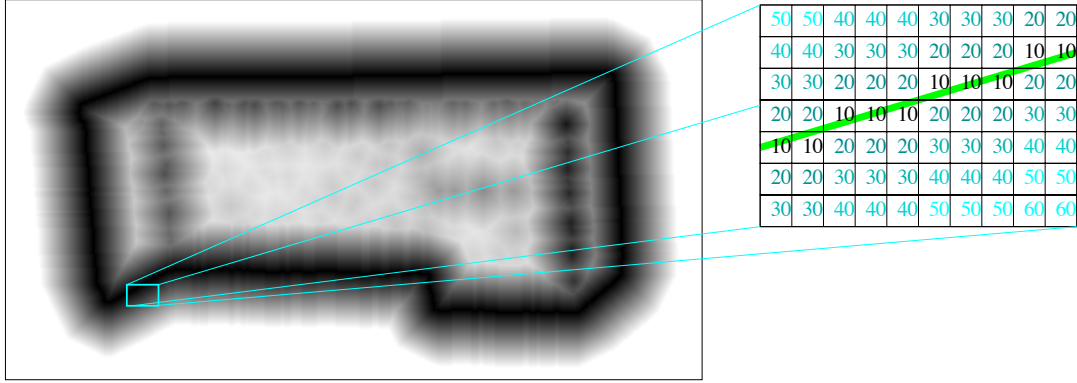


Abbildung 4.3: Das pixelbasierte Distanzmaß: Der Ausschnitt stellt zur Veranschaulichung eine idealisierte Kante mit konstanter Ausprägung dar, welche im Bild so nicht gegeben ist.

Gradientenstärke größer als eine Minimalschwelle, so wird rekursiv die Nachbarschaft dieser Position untersucht, wobei der Abstand zum Ausgangspunkt vermerkt bleibt. So wird entsprechend der Definition aus Gleichung 4.1 für jeden Punkt der Nachbarschaft des Kantenpunktes das Distanzmaß berechnet. Da im Kanten-Distanz-Bild immer nur die Distanz zur nächsten Kante berücksichtigt wird, muss darüberhinaus geprüft werden, ob in diesem bereits durch einen anderen Kantenpixel eine kleinere Distanz eingetragen worden ist. Es wird also jeweils der nach Gleichung 4.1 minimale Abstand

$$d(x, y) = \min_K d_K(x, y) \quad (4.2)$$

an der Stelle  $(x, y)$  im Kanten-Distanz-Bild eingetragen.

Die Berechnung der Abstände einzelner Pixel der Nachbarschaft wird aus Gründen der besseren Effizienz nicht exakt berechnet. Stattdessen erfolgt eine Approximierung von horizontalen und vertikalen Abständen zwischen zwei Pixeln durch  $2 \cdot 1 = 2$  und diagonalen durch  $2 \cdot \sqrt{1^2 + 1^2} = 2 \cdot \sqrt{2} \approx 3$ . Abbildung 4.3 zeigt das Ergebnis der Anwendung des Algorithmus am Beispiel des schon in Abbildung 4.2 dargestellten Schreibtisches.

Durch die Analyse der Nachbarschaft eines Kantenpunktes ist es darüber hinaus möglich, die Richtung  $\gamma$  der jeweils im Sinne des Abstandsmaßes nächstliegenden

Kante durch

$$\begin{aligned}
 d_h &\rightarrow \text{Abstand in horizontaler Richtung vom Kantenpunkt} \\
 d_v &\rightarrow \text{Abstand in vertikaler Richtung vom Kantenpunkt} \\
 \gamma &= \text{atan2}(d_h, d_v)
 \end{aligned} \tag{4.3}$$

zu bestimmen. Diese Richtung wird im Abschnitt 4.4 für die Projektionsoptimierung benötigt.

Um die Distanz einer Modellkante  $E_M$  zu ermitteln, erfolgt die Addition der von ihr überstrichenen Pixel des Kanten-Distanz-Bildes. So ist ein Maß

$$D(E_M) = \sum_{(x,y) \in E_M} d(x, y) \tag{4.4}$$

definiert, dass die in 2.6 gestellten Bedingungen erfüllt.

## 4.3 Projektion

Da in dieser Arbeit der Vergleich zwischen Bild und Modell durch das zweidimensionale Kanten-Distanz-Bild realisiert wird, ist es erforderlich, das Modell in diesen Bildraum abzubilden. Dies geschieht mittels einer *Zentralprojektion* welche von der vereinfachenden Annahme einer Lochkamera ausgeht. Von den Kamera-parametern wird also nur die Brennweite  $f$  benötigt, nicht jedoch weitere zu kalibrierende Parameter, wie sie durch verschiedene Linsen-Verzeichnungen [Jäh97] auftreten können. Da anschließend an die initiale Projektion diese noch optimiert wird, werden zumindest lineare Abbildungsfehler automatisch korrigiert, so dass von diesem einfachen Kameramodell ausgegangen werden kann.

Vor der eigentlichen Projektion erfolgt eine Transformation in das Kamerakoordinatensystem  $C$ , welche basierend auf der Lage- und Positionshypothese durchgeführt wird. Im Kamerakoordinatensystem befindet sich die Kamera im Ursprung und ihre optische Achse fällt mit der dritten Raumachse  $z_c$  zusammen (siehe Abbildung 4.4). Alle Berechnungen werden in *homogenen Koordinaten* durchgeführt, um Translationen  $T$  und Rotationen  $R$  der Modellkoordinaten zusammen



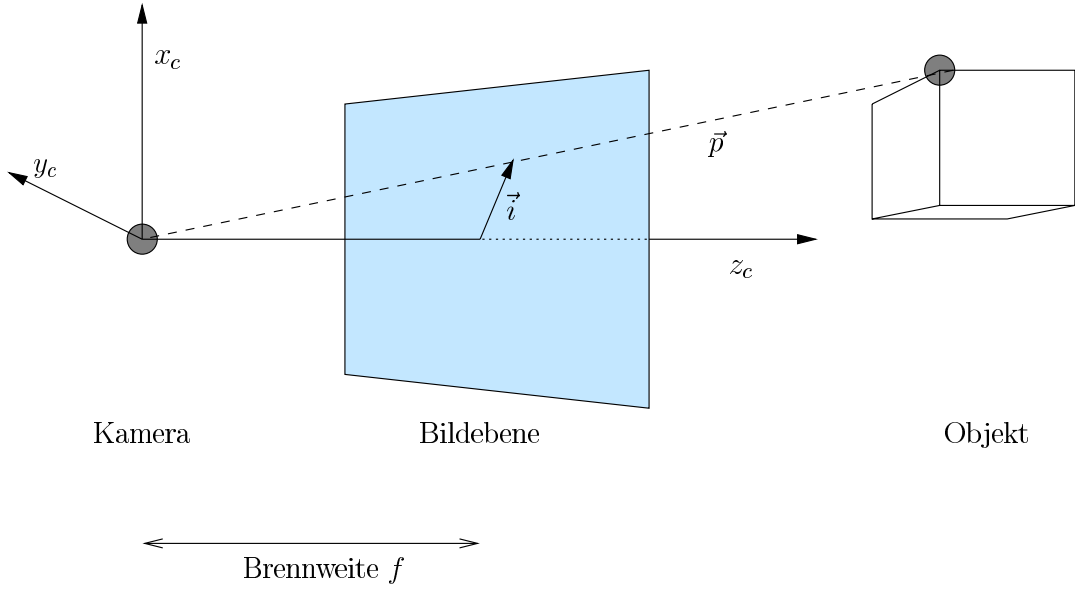


Abbildung 4.4: Lage des Kamerakoordinatensystems und Projektion eines Punktes  $\vec{p}$  auf die Bildkoordinaten  $\vec{i}$

darstellen zu können. Ausführlichere Erläuterungen zu homogenen Koordinaten und Transformationen sind im Anhang A ab Seite 53 zu finden. Die Transformation eines in homogenen Koordinaten gegebenen Modellpunktes  $\vec{m}$  wird durch die lineare Abbildung

$$\begin{aligned}\vec{p} &= T(\vec{t}) \cdot R(\alpha, \beta, \gamma) \cdot \vec{m} \\ &= H \cdot \vec{m}\end{aligned}\tag{4.5}$$

realisiert. Der Aufbau der  $4 \times 4$ -Matrix  $H$  kann Gleichung A.5 entnommen werden. Sie berechnet zuerst eine Rotation des Objektes um die Winkel  $\alpha$ ,  $\beta$ ,  $\gamma$  um die drei Achsen des Kamerakoordinatensystems und eine anschließende Translation  $\vec{t}$ . Zur Abbildung einer Kamerakoordinate  $\vec{p}$  in die Bildkoordinate  $\vec{i}$  wird die schon vorgestellte Zentralprojektion verwendet [Jäh97, Seite 193ff]:

$$\begin{aligned}\vec{i} &= \text{Proj}(\vec{p}) \\ \begin{pmatrix} i_1 \\ i_2 \end{pmatrix} &= f \begin{pmatrix} \frac{p_1}{p_3} \\ \frac{p_2}{p_3} \end{pmatrix}\end{aligned}\tag{4.6}$$

## 4.4 Optimierung

Wenn keine transformationsinvarianten Merkmale verwendet werden, ist es zur Klassifikation notwendig, die Lage des Objektes im Bild vor der Klassifikation zu ermitteln, da ansonsten keine robuste Erkennung möglich ist. Ein allgemeines Objekt unterscheidet sich aus unterschiedlichen Perspektiven signifikant, so dass eine Unterscheidung verschiedener Objekte unmöglich wird. Aus diesem Grunde wird basierend auf dem definierten Abstandmaß 4.2 eine Projektionsoptimierung durchgeführt, welche von einer initialen Transformationshypothese ausgeht und durch eine iterativ angewendete lineare Regression realisiert wird.

Der Algorithmus ist hierbei zweigeteilt, wie es auch schon in Abbildung 2.2 dargestellt ist. Auf Basis der aktuellen Transformationshypothese wird eine Projektion des Modells durchgeführt, wobei die Sichtbarkeitsregeln für die Auswahl der zu projizierenden Kanten sorgen. Anschließend wird eine diskrete Abtastung der projizierten Modellmerkmale vorgenommen (z.B. werden je Merkmal 20 Punkte betrachtet). An diesen Abtastkoordinaten  $\vec{p}_i$  wird jeweils die pixelweise Distanz zum Bild durch das Kantendistanzmaß 4.2 ermittelt. Da bei diesem Maß auch die Richtung zum nächstgelegenen Kantenpixel bekannt ist, kann ein Vektor berechnet werden, der angibt, wohin der abgetastete Punkt verschoben werden muss, um die Distanz zu minimieren. Diese Vektoren an einer Kante sind sozusagen „Kraftlinien“, die an mehreren Stellen am Modell ziehen, und so die Veränderung der Projektion festlegen. Auf diese Weise ist eine Menge von Ausgangspunkten  $\vec{p}_i$  und, durch die Verschiebungsvektoren, eine Menge von Zielpunkten  $\vec{q}_i$  gegeben, welche jedoch nur eine Transformation innerhalb der Bildebene wiedergeben können. Die Tiefeninformation kann nicht direkt gewonnen werden, was bedeutet, dass die dritte Koordinate der Punkte  $\vec{q}_i$  vorerst von  $\vec{p}_i$  übernommen werden muss. Auf Basis dieser beiden Punktmengen kann eine *multidimensionale lineare Regression* durchgeführt werden, welche eine Abbildungsmatrix berechnet, die die gegebene Verschiebung von Punkten am besten (also mit geringstem Abweichungsfehler) approximiert. Durch die Beibehaltung der Tiefeninformation kann eine Optimierung der Transformation parallel zur optischen Achse der Kamera nicht direkt erfolgen. Die lineare Optimierung berechnet aber eine optische Verzerrung des Modells, aus der diese Information gemäß dem in Abschnitt 4.4.3

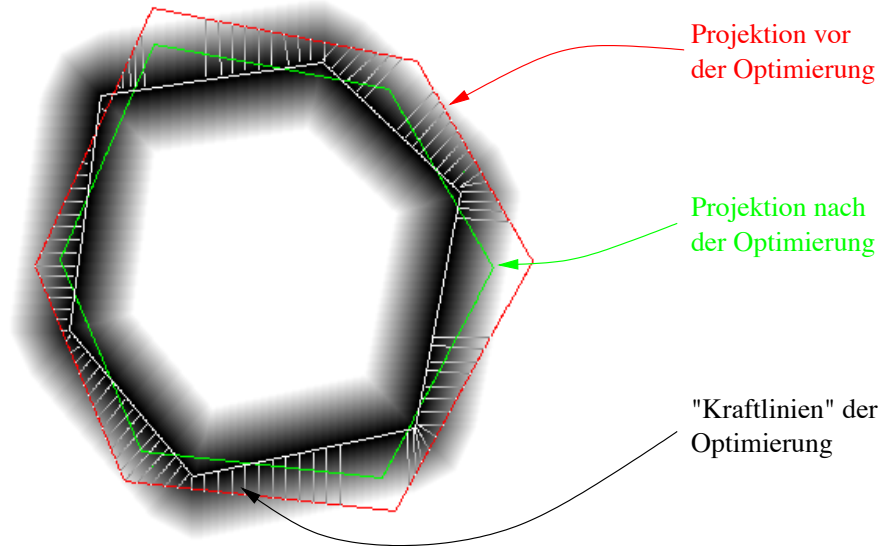


Abbildung 4.5: Ein Schritt der iterativen Projektionsoptimierung. Das Kantendistanz-Bild ist hier idealisiert dargestellt

vorgestellten Verfahren, ermittelt wird. Die Optimierung durch Anwendung der Regression wird im folgenden Abschnitt erläutert.

#### 4.4.1 Multidimensionale lineare Regression homogener Transformationsmatrizen

Gegeben ist eine Folge von Ortsvektoren  $\vec{p}_i$  in homogener Koordinatendarstellung und ihre bekannten korrespondierenden Transformationen  $\vec{q}_i$ , die jeweils beide in allgemeiner Lage, also nicht koplanar, liegen. Durch die multidimensionale lineare Regression wird nun die Abbildung  $\hat{H}$  ermittelt, die das quadratische Fehlermaß

$$E = \sum_i \left( \vec{q}_i - \hat{H} \cdot \vec{p}_i \right)^2 \quad (4.7)$$

minimiert, weshalb das Verfahren auch als *Methode der kleinsten Quadrate* bekannt ist. Die beste Lösung für  $\hat{H}$  kann nach [Kre98, Seite 166ff] durch folgende

Gleichung bestimmt werden:

$$\begin{aligned}
 & \overbrace{\left[ \begin{array}{ccc|c} \sum_i p_{1i}^2 & \sum_i p_{1i}p_{2i} & \sum_i p_{1i}p_{3i} & \sum_i p_{1i} \\ \sum_i p_{1i}p_{2i} & \sum_i p_{2i}^2 & \sum_i p_{2i}p_{3i} & \sum_i p_{2i} \\ \sum_i p_{1i}p_{3i} & \sum_i p_{2i}p_{3i} & \sum_i p_{3i}^2 & \sum_i p_{3i} \\ \hline \sum_i p_{1i} & \sum_i p_{2i} & \sum_i p_{3i} & 1 \end{array} \right]}^A \quad \overbrace{\left[ \begin{array}{ccc|c} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ \hline S_1 & S_2 & S_3 & 1 \end{array} \right]}^{\text{gesuchte Koeffizienten } \hat{H}} \quad (4.8) \\
 & = \underbrace{\left[ \begin{array}{ccc|c} \sum_i q_{1i}p_{1i} & \sum_i q_{2i}p_{1i} & \sum_i q_{3i}p_{1i} & \sum_i p_{1i} \\ \sum_i q_{1i}p_{2i} & \sum_i q_{2i}p_{2i} & \sum_i q_{3i}p_{2i} & \sum_i p_{2i} \\ \sum_i q_{1i}p_{3i} & \sum_i q_{2i}p_{3i} & \sum_i q_{3i}p_{3i} & \sum_i p_{3i} \\ \hline \sum_i q_{1i} & \sum_i q_{2i} & \sum_i q_{3i} & 1 \end{array} \right]}_B
 \end{aligned}$$

Die in der Darstellung der Matrizen  $A$  und  $B$  durch Striche abgetrennten letzten Zeilen und Spalten ergeben sich auf Grund der homogenen Koordinaten der Vektoren  $\vec{p}_i$  und  $\vec{q}_i$ , bei denen die vierte Komponente immer 1 ist. Das lineare Gleichungssystem kann durch Standardverfahren wie Cholesky-Zerlegung oder Gaußelimination (beide in [Sto93]) gelöst werden:

$$\begin{aligned}
 A \hat{H} &= B \\
 \hat{H} &= A^{-1} B
 \end{aligned} \quad (4.9)$$

#### 4.4.2 Iteration

Durch  $\hat{H}$  ist eine Änderungshypothese für die aktuelle Transformation  $H$  gegeben. Durch

$$\bar{H} = \hat{H}H \quad (4.10)$$

erfolgt die Berechnung der optimierten Transformation  $\bar{H}$ . Um während der Iterationen bereits eine Translation des Objektes entlang der optischen Achse, die von der initialen Hypothese abweicht, zu optimieren, wird die in der Matrix  $\bar{H}$  enthaltene Skalierungsinformation ausgewertet. Aus der Projektionsgleichung 4.6 ist ersichtlich, dass eine Skalierung  $S$  eines starren Objektes gemäß Gleichung A.4

in der Abbildung bei konstanter Brennweite  $f$  durch eine Translation entlang der optischen Achse  $z_c$  begründet ist:

$$s \begin{pmatrix} i_1 \\ i_2 \end{pmatrix} = f \begin{pmatrix} \frac{p_1}{p_3/s} \\ \frac{p_2}{p_3/s} \end{pmatrix} = f \begin{pmatrix} p_1 \frac{s}{p_3} \\ p_2 \frac{s}{p_3} \end{pmatrix} \quad (4.11)$$

Daher kann ausgehend von der Beziehung [Jän98, Seite 147ff]

$$\begin{aligned} \det(S \cdot \bar{H}) &= \det(S) \cdot \det(\bar{H}) \\ &= (s_1 s_2 s_3) \cdot \det(\bar{H}) \\ &= s^3 \cdot \det(\bar{H}) \quad (\text{für } s_1 = s_2 = s_3 = s) \end{aligned} \quad (4.12)$$

aus der Determinante der Transformationsmatrix  $\bar{H}$  direkt auf die Translation geschlossen werden. Die Transformationsmatrix  $H'$ , die diese Änderung berücksichtigt, gleicht in fast allen Komponenten der Matrix  $\bar{H}$ . Lediglich  $H'_{34}$  wird gemäß

$$H'_{34} = \frac{\bar{H}_{34}}{\sqrt[3]{\det(\bar{H})}} \quad (4.13)$$

gesetzt. Mit der Transformation  $H'$  wird das Verfahren dann iteriert, bis nur noch sehr kleine Veränderungen zwischen zwei Durchläufen auftreten, d.h.  $\hat{H}$  annähernd der Einheitsmatrix entspricht oder eine festgelegte Anzahl Iterationen durchgeführt wurde.

### 4.4.3 Lagerekonstruktion

Nachdem die Berechnung einer optimierten Transformation unter Beibehaltung der tiefenkodierenden Komponenten der Punkte durch die lineare Regression erfolgt ist, kann durch die Analyse der Scherungen und Verzerrungen, die die lineare Regression für die Transformationsmatrix  $H'$  ermittelt, die Lageinformation approximiert werden. Dies ist möglich, wenn die betrachteten Objekte eine relativ geringe Tiefe im Vergleich zur Brennweite der Kamera haben. Ist dies nicht gegeben, so ist eine Verschlechterung der Qualität der Schätzung zu erwarten, da durch die lineare Abbildung perspektivische Verzerrungen nicht erfasst werden können.

Wie bereits angesprochen, kann eine Rotation parallel zur optischen Achse  $z_c$  der Kamera sowie eine Translation direkt durch die lineare Regression ermittelt werden. Die Berechnung des Rotationswinkels  $\gamma$  um die  $z_c$ -Achse kann gemäß Gleichung A.6 aus der optimierten Transformationsmatrix  $H'$  errechnet werden. Die Translation  $\vec{t}$  ist direkt aus der vierten Spalte von  $H'$  ablesbar. Es soll nun aus den verbleibenden Komponenten der Matrix auf die Rotationswinkel um die  $x_z$  und die  $y_c$ -Achse geschlossen werden. Durch Kenntnis der Matrix  $Z$  (siehe auch Gleichung A.3)

$$Z = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 & t_1 \\ \sin \gamma & \cos \gamma & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (4.14)$$

die nur die beiden bekannten Teil-Transformationen kodiert, kann eine reduzierte Matrix  $W$

$$W = Z^{-1} \cdot H' \quad (4.15)$$

berechnet werden, welche die verbleibenden Transformationen enthält. Unter der Voraussetzung, dass  $H'$  nur durch Rotationen und Translationen des Modells erklärt werden kann, kann  $W$  gemäß Gleichung A.5 auch als

$$W = \begin{pmatrix} \cos \beta & \sin \alpha \sin \beta & \cos \alpha \sin \beta & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ -\sin \beta & \cos \beta \sin \alpha & \cos \beta \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.16)$$

geschrieben werden, wobei die Winkel der Rotationen um die  $x_c$  und die  $y_c$ -Achse mit  $\alpha$  und  $\beta$  bezeichnet sind. Die Voraussetzung ist für ein Bild, das tatsächlich das zu prüfende Objekt zeigt, erfüllt, da die Objekte als starr betrachtet werden.

Durch das Fehlen der Tiefeninformation während der linearen Regression ist die Matrix  $W$  jedoch nicht vollständig gegeben, da sich für die dritte Zeile der Matrix  $\hat{H} = (0, 0, 1, 0)$  eine Identitätsabbildung der Tiefenkoordinaten der Punkte ergibt. Eine Berechnung der Winkel  $\alpha$  und  $\beta$  durch die Komponenten der dritten Zeile von  $W$  ist also nicht möglich, da diese nicht durch die lineare Regression

ermittelt wird. Dennoch ist die Berechnung der Rotationswinkel durch

$$\begin{aligned}\alpha &= \text{atan2}(W_{12}, W_{13}) \\ \beta &= \text{atan2}\left(\sqrt{1 - W_{11}^2} \cdot \text{sgn}(-W_{12} \cdot W_{23}), W_{11}\right)\end{aligned}\quad (4.17)$$

möglich. Für  $\beta = \pm \frac{\pi}{2}$  ergibt sich, wie in Gleichung A.7 diskutiert, eine Singularität. Die relativ komplizierte Gleichung für  $\beta$  resultiert aus der Definition der Funktion  $\text{atan2}$  in Gleichung A.8, welche die Vorzeichen der Parameter auswertet. Auf diese Weise ist eine vollständige Schätzung der sechs Lageparameter des Objektes gegeben.

## 4.5 Klassifikation

Die Klassifikation verschiedener Objekte ist durch das Distanzmaß aus Gleichung 4.4 leicht zu realisieren. Für jedes Modell  $m$  wird die normierte Distanz

$$d_m = \frac{1}{N_m} \sum_{n=1}^{N_m} d(E_n) \quad (4.18)$$

über alle  $N_m$  Kanten  $E_n$  errechnet, um mehrere Objekte vergleichen zu können. Als Ergebnis der Klassifikation wird das Modell  $m_{opt}$  geliefert, dass nach der Transformationsoptimierung den geringsten Abstand zum Bild hat:

$$m_{opt} = \underset{m}{\text{argmin}} d_m \quad (4.19)$$

## 4.6 Zusammenfassung

Das hier vorgestellte Verfahren zur Objekterkennung auf Basis der Konturkanten der Modelle bietet die Möglichkeit aus zweidimensionalen Intensitätsbildern sowohl ein Klassifikationsergebnis als auch eine Positions- und Lageinformation des betrachteten Objektes zu erlangen. Das vorgestellte Distanzmaß und die Optimierung durch lineare Regression sind Ansätze, dieses zu realisieren. In Kapitel 6 werden die Fähigkeiten und Grenzen dieser Umsetzung betrachtet.





# Kapitel 5

## Realisierung

Nachdem die Umsetzung einer modellbasierten Objekterkennung durch Verwendung von Konturkanten erklärt worden ist, soll in diesem Abschnitt auf die Realisierung des Systems aus softwaretechnischer Sicht eingegangen werden. Diese Arbeit umfasst mit der automatischen Aufbereitung und der Erkennung eigentlich zwei Programme, welche hier aber zusammen erläutert werden. Ersteres erzeugt eine Menge von Identifikationsmerkmalen mit zugehörigen Parametern und Sichtbarkeitsregeln. Diese Daten werden für alle bekannten Objekte generiert und dann vom Erkennungsprogramm verwendet.

### 5.1 Anforderungen

In den vorangegangenen Kapiteln ist bereits mehrfach davon die Rede gewesen, dass in dieser Arbeit keine Spezialisierung der Erkennung auf die Auswertung der Konturkanten gegeben sein soll. Vielmehr wird bei der Realisierung des Projektes auf eine modulare Architektur der verwendeten Merkmale geachtet, um sowohl die Modellaufbereitung als auch die Erkennung erweiterbar zu gestalten und Funktionen für eine Experimentierumgebung bereitzustellen. Sowohl die automatische Modellaufbereitung als auch die eigentliche Objekterkennung unterliegen diesen Anforderungen. Auf Grund der engen Verknüpfung beider Teile sind auch bei

der Softwarearchitektur diese Parallelen auszunutzen, um doppelte Implementierungen zu vermeiden. Dennoch soll jedes Programm voneinander unabhängig verwendet werden können.

## 5.2 Software-Architektur

Aus den zuvor genannten Gründen wurde als Softwarearchitektur ein objektorientierter Ansatz gewählt, welcher in Abbildung 5.1 als Klassenübersichtsdiagramm veranschaulicht ist. Als virtuelle Klassen dienen `Model`, `IdentFeature` und `Rule` der Abstraktion der konkreten Implementierung der Erkennung. Der vorgestellte Ansatz einer Erkennung auf Basis der Konturkanten als Identifikationsmerkmal ist somit nur eine dieser speziellen Realisierungen. Sie ist in der Klassenübersicht durch `ModelEdge`, `ChamferEdge`, sowie die beiden Regelklassen `VisibilityCone` und `Outline` dargestellt und als *Modellrepräsentation* bezeichnet.

Die Klasse `Camera` beinhaltet Methoden zur Projektion und Transformation auf Basis eines Kameramodells und kann auch weitere Parameter der Kamerakalibrierung berücksichtigen. Das in Kapitel 4.2 vorgestellte Kantendistanzbild ist durch die Klasse `ChamferEdge` realisiert. Die für die Projektionsoptimierung benötigte lineare Regression ist in der Klasse `Regression` realisiert und kann auch losgelöst von diesem System in anderen Aufgaben eingesetzt werden. Von zentraler Bedeutung sind die Klassen `Preparation` und `Recognition`, da sie die beiden benötigten Programme zur Modellvorbereitung und Erkennung realisieren. Wie aber ersichtlich ist, verwenden sie größtenteils gleiche Klassen, um Änderungen und Erweiterungen in beiden Teilen transparent halten zu können.

Die konkrete Implementierung geschieht in `C++`, da hier hohe Verarbeitungsgeschwindigkeit für die zeitkritische Erkennung mit einer objektorientierten Programmierung einhergehen. Mit der Bibliothek `LEDA` [MN00] steht darüberhinaus eine umfangreiche Algorithmensammlung zur Verfügung, die die Realisierung der Vektorengeometrie und linearen Algebra dieser Arbeit vereinfacht.

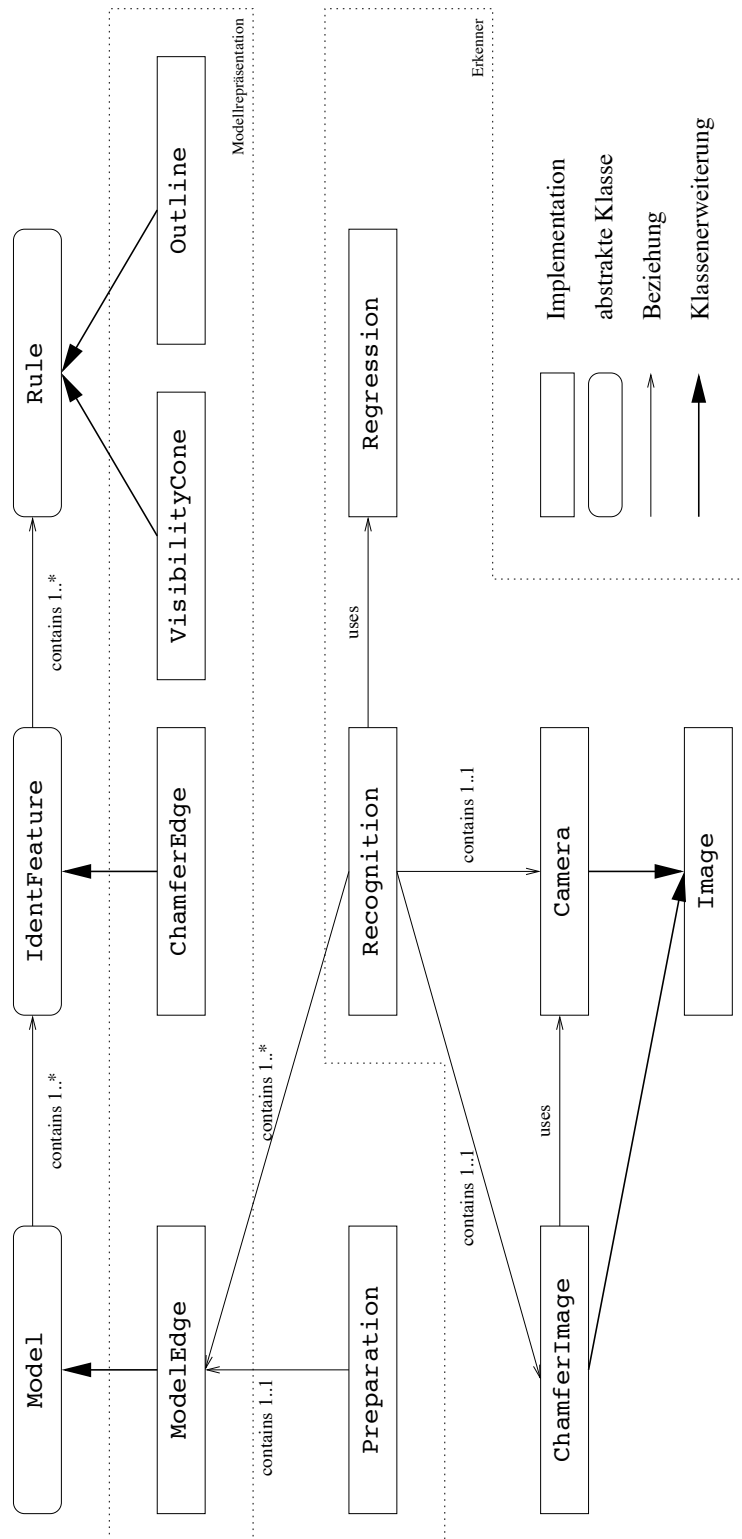


Abbildung 5.1: Klassenübersicht



# Kapitel 6

## Auswertung

Die Auswertung der verwendeten Ansätze soll in diesem Kapitel erfolgen. Hierzu werden die automatische Modellvorbereitung und die Erkennung getrennt betrachtet, da sie auch unabhängig voneinander eingesetzt werden. Die Anzahl der zur Verfügung stehenden Modelle ist für eine umfassende Bewertung der Verfahren zur Zeit noch zu gering, aber die prinzipielle Funktionalität und die Probleme lassen sich erkennen.

### 6.1 Automatische Modellvorbereitung

Für die Erkennung auf Basis der Konturkanten eines Objektes ist wichtig, dass das Objekt aus jeder Perspektive durch entsprechende Modellkanten, die in diesem Ansatz die Identifikationsmerkmale darstellen, begrenzt ist. Dies ist bei diesen einfachen Merkmalen leicht durch Visualisierung der Modelle mit den, entsprechend des in Kapitel 3 vorgestellten Verfahrens, generierten Kantenmerkmalen möglich. Daher werden nun einige Abbildungen von Modellen mit den erzeugten Merkmalen aus verschiedenen Domänen präsentiert. So finden sich darunter zum Beispiel Möbel und Einrichtungsgegenstände, aber auch ein Objekt, das für die Bildverarbeitung in der industriellen Fertigung interessant sein kann. Zur besseren Sichtbarkeit sind die jeweiligen Endpunkte der Kanten mit einem kleinen Würfel visualisiert. Deutlich erkennbar ist am Beispiel des Objektes „Mülle-

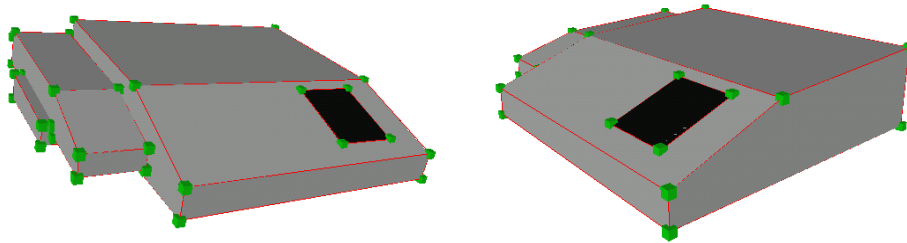


Abbildung 6.1: Ein einfaches Faxgerät aus zwei Perspektiven

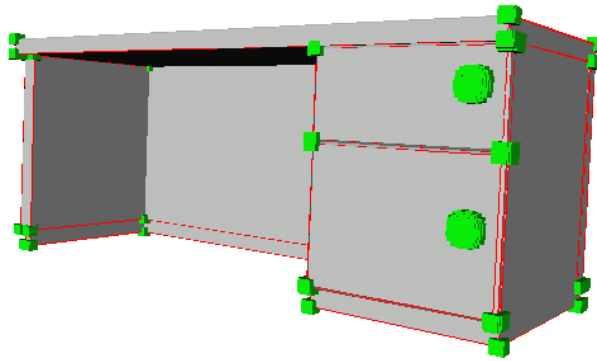


Abbildung 6.2: Ein Bürotisch

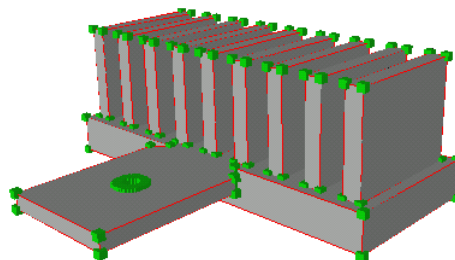


Abbildung 6.3: passiver Kühler für integrierte Schaltkreise

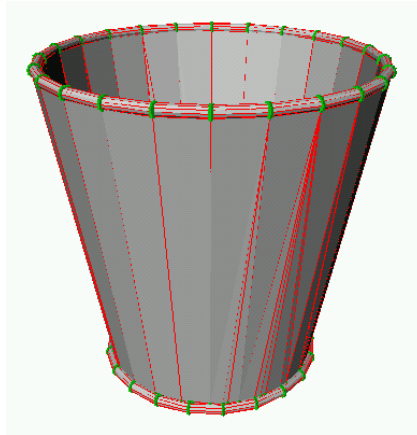


Abbildung 6.4: Ein typischer Mülleimer

mer“ (Abbildung 6.4) das Problem der Aufbereitung von Objekten mit runden Elementen. Die in Abschnitt 3.2 angeführte Approximation hat eine große Anzahl von Identifikationsmerkmalen zur Folge. Eine Erkennung ist zwar dennoch möglich, aber es werden bei der Projektion eines solchen Modells sehr viele Regeln geprüft, obwohl nur eine geringe Anzahl der Kanten überhaupt als Kontur in Frage kommt.

## 6.2 Erkennung

Die Evaluation der Erkennung und der Lageschätzung erfolgt nur auf künstlichen Bildern, da die verwendeten Modelle ausschließlich aus freien Quellen im Internet bezogen wurden und somit die zu Grunde liegenden realen Objekte nicht zur Verfügung stehen. Im Anhang B sind einige der verwendeten künstlich erzeugten Bilder dargestellt. Diese haben allerdings den Vorteil, dass mit ihnen idealisierte Bedingungen geschaffen werden können, mit denen die Eigenschaften des Erkennungsalgorithmus gemessen werden können. So ist zum Beispiel die genaue Lage des Objektes im Bild bekannt und es kann die grundsätzliche Leistungsfähigkeit der Projektionsoptimierung getestet werden.

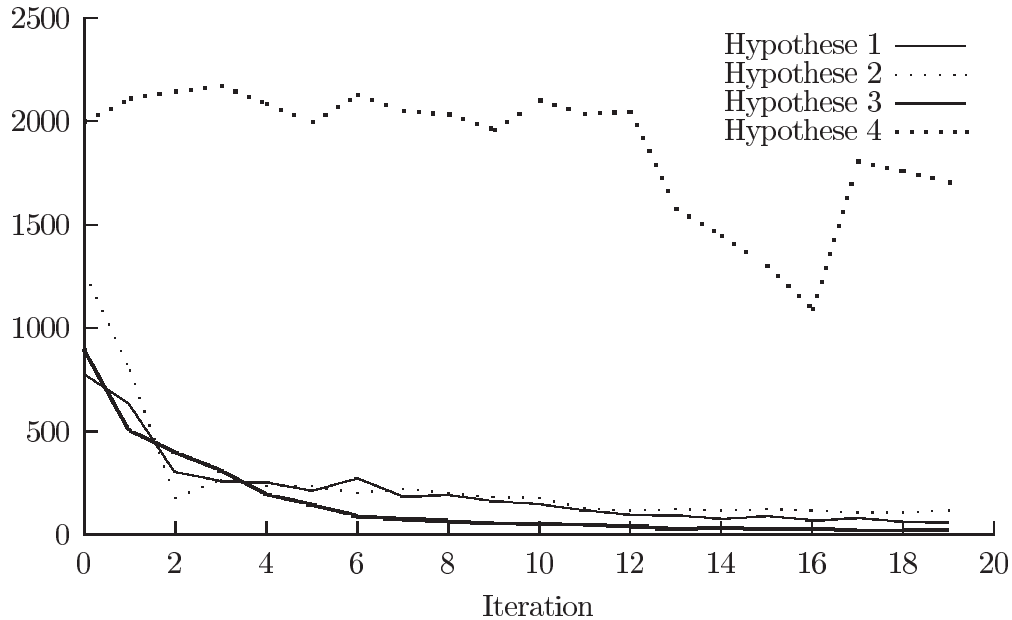


Abbildung 6.5: Distanz des Objektes „Fax“ während der Projektionsoptimierung

### 6.2.1 Projektionsoptimierung und Lagehypothese

Die Qualität der initialen Lagehypothese ist von entscheidender Bedeutung für das Klassifikationsergebnis. Die Optimierung der Projektion erfolgt ausgehend von dieser Hypothese. Es ist daher wichtig, zu wissen, wie gut die Lage des Objektes vorhergesagt werden muss, um noch zu korrekten Klassifikationsergebnissen zu gelangen. Dies ist jedoch stark abhängig von der Art der Abweichung von der realen Lage und den geometrischen Gegebenheiten des Modells, so dass eine allgemeine Auswertung hier nicht möglich ist.

Der Vorgang der Projektionsoptimierung lässt sich jedoch am Verlauf der jeweiligen Distanz nach Gleichung 4.18 des Modells zum aktuellen Bild veranschaulichen. Abbildung 6.5 zeigt den Verlauf der Distanz über 20 Iterationen am Beispiel des Modells „Fax“ bei vier unterschiedlichen initialen Hypothesen, die durch ein 6-Tupel  $(T_1, T_2, T_3, \alpha, \beta, \gamma)$  gemäß Gleichung A.5 gegeben sind. Als Test-Bild dient ein künstlich erzeugtes Bild, welches im Anhang B zu sehen ist. Es wurde mit den Transformationsparametern  $(0, 0, 10, 110, 30, 0)$  erzeugt, die die optimale



	initiale Hypothese	ermittelte Lage
korrekte Lage	(0, 0, 10, 110, 30, 0)	—
Hypothese 1	(0, 0, 10, 110, 10, 0)	(0.0, 0.0, 10.0, 110.2, 29.2, -0.2)
Hypothese 2	(0, 0, 12, 130, 30, 0)	(0.0, 0.0, 9.9, 110.3, 32.3, 0.7)
Hypothese 3	(0, 0, 10, 110, 30, 20)	(0.0, 0.0, 9.9, 110.0, 30.1, 0.1)
Hypothese 4	(0, 0, 10, 0, 0, 0)	keine Konvergenz

Tabelle 6.1: Ergebnisse der Projektionsoptimierung des Objektes „Fax“

Lösung darstellen. Die Tabelle 6.1 gibt die initialen Hypothesen der Optimierungen sowie die ermittelte Lage nach 20 Iterationen an. Es ist deutlich sichtbar, dass geringe Abweichungen von der korrekten Initialisierung (Hypothese 1–3) durch die Optimierung ausgeglichen werden können. Liegt jedoch wie im Falle der Hypothese 4 eine große Fehleinschätzung vor, so kann die Projektionsoptimierung dies nicht ausgleichen.

Die durch die Projektionsoptimierung gewonnene Lagehypothese ist in ihrer Qualität stark abhängig von der Art und Lage des Objektes im Bild. Insbesondere die Tiefenschätzung aus Abschnitt 4.4.3 kann nur dann zu vernünftigen Ergebnissen kommen, wenn nicht nur koplanare Merkmale in der aktuellen Projektion sichtbar sind. Am Beispiel der Optimierung des Modells „Fax“, ausgehend von Hypothese 3 in Tabelle 6.1, zeigt Abbildung 6.6 die absolute Abweichung der drei Orientierungswinkel  $\alpha$ ,  $\beta$  und  $\gamma$  gegenüber den korrekten Werten. Die Konvergenz der Abweichungen gegen Null ist erkennbar.

Der Vorgang der Optimierung kann darüberhinaus visualisiert werden (siehe Abbildung 6.7). In das Kanten-Distanz-Bild sind die jeweils sichtbaren Identifikationsmerkmale des Modells, basierend auf der aktuellen Hypothese, abgebildet und zusätzlich die „Kraftlinien“ der Optimierung aus Abschnitt 4.4 eingeblendet.

### 6.2.2 Klassifikation

Die Klassifikation eines Bildes wird durch die Berechnung der Distanzen der voneinander unabhängig optimierten, gegebenen Modelle realisiert. Da zu jeder Iteration die Distanz anhand des Abstandsmaßes ermittelt wird, kann jeweils die

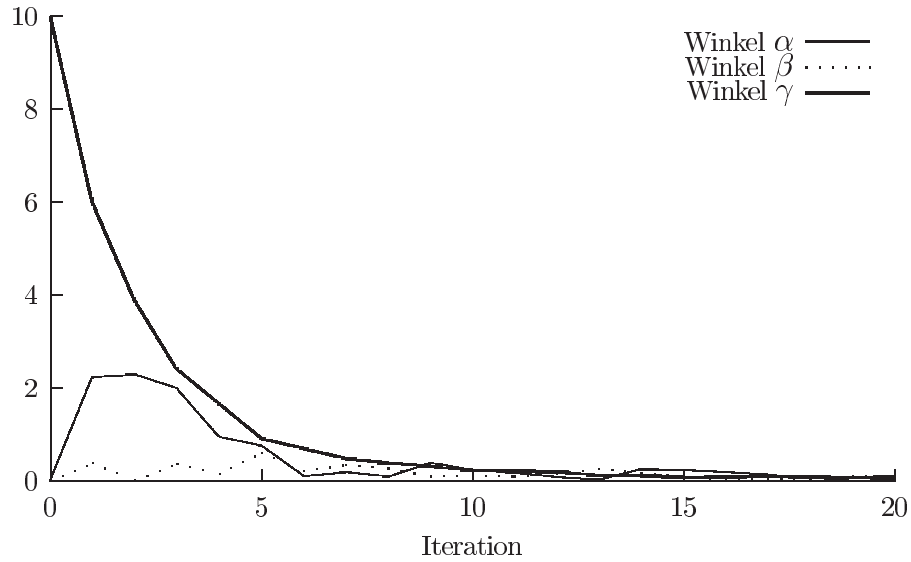


Abbildung 6.6: Verlauf der Abweichung von der korrekten Lage für alle drei Raumwinkel

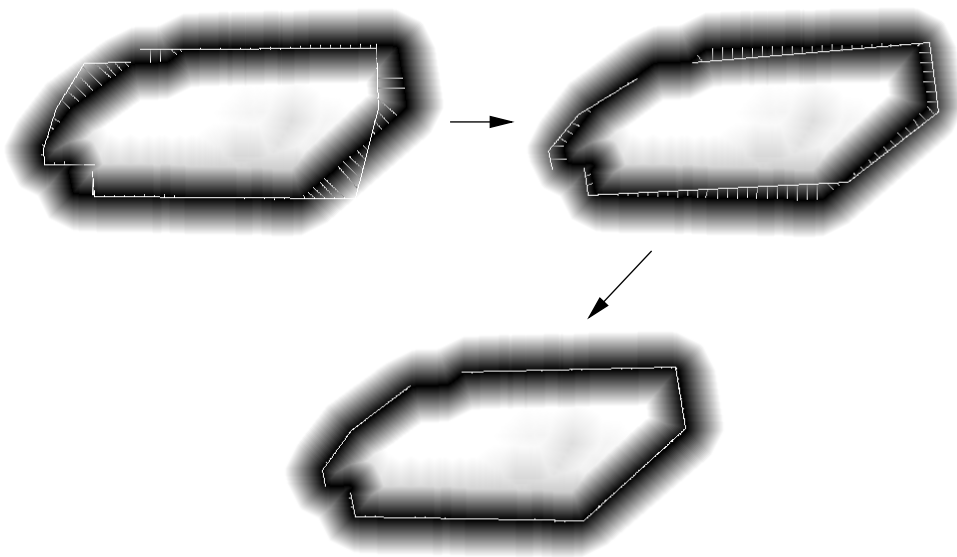


Abbildung 6.7: Visualisierung der Iterationen 1, 2 und 20 der Projektionsoptimierung ausgehend von Hypothese 1

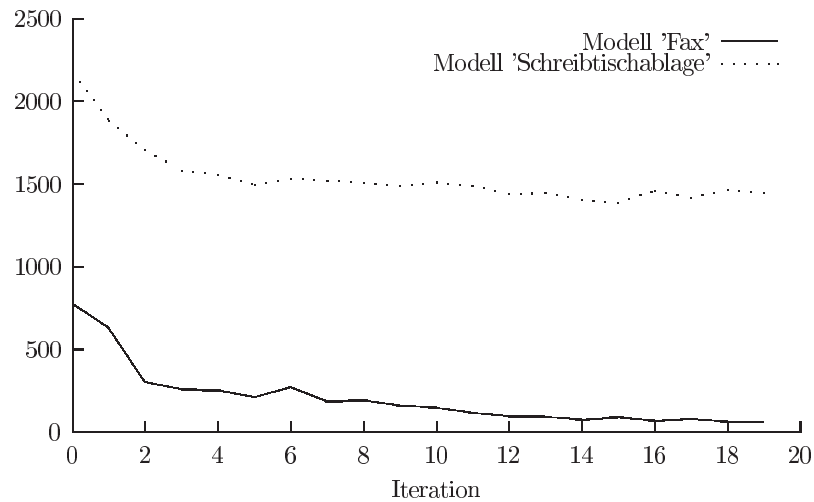


Abbildung 6.8: Klassifikation I – Verlauf der Distanzen der Modelle „Fax“ und „Schreibtischablage“ bei einem Bild, das „Fax“ enthält

Berechnung einer aktuellen Klassifikationshypothese nach Gleichung 4.19 erfolgen. In Abbildung 6.8 ist die Entwicklung der Distanz des Modells „Fax“, welches tatsächlich im Bild zu sehen ist, und des Modells „Schreibtischablage“ zu sehen. Hier ist der Unterschied in der Distanz deutlich zu erkennen. Eine Klassifikation durch Auswahl des Modells mit dem geringsten Abstand liefert die korrekte Lösung („Fax“). Den Verlauf einer weiteren Klassifikation zeigt Abbildung 6.9. In diesem Falle wird das Bild der „Diskettenbox“ aus Anhang B klassifiziert, wobei die Erkennung mit den Modellen „Schreibtischablage“, „Fax“ und „Diskettenbox“ durchgeführt wird. Auch in diesem Versuch ist das Ergebnis klar abzulesen. Weiterhin ist die Konvergenz gegen eine optimale Ausrichtung – auch der beiden nicht passenden Modelle – zu sehen. Bei beiden Experimenten wird ausgehend von einer initialen Hypothese, die in den drei rotatorischen Freiheitsgraden  $25^\circ$  Abweichung von der Ideallage hat, optimiert.

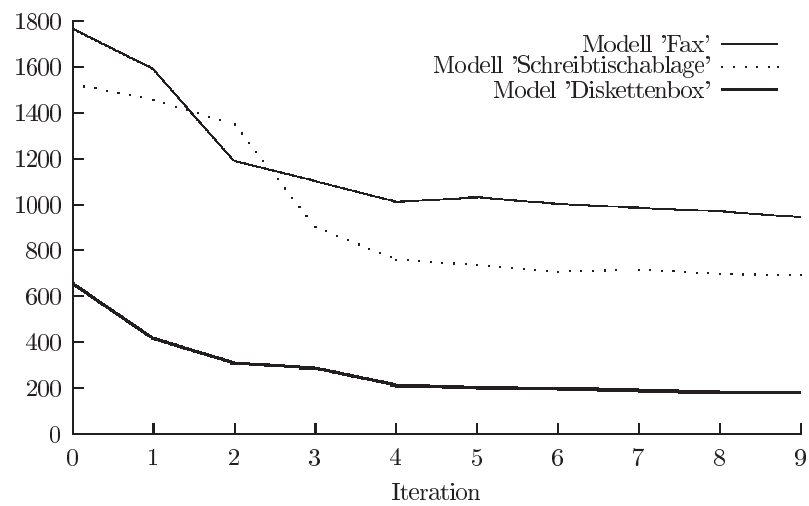


Abbildung 6.9: Klassifikation II – Verlauf der Distanzen der Modelle „Fax“, „Schreibtischablage“ und „Diskettenbox“ bei einem Bild, das „Diskettenbox“ enthält

# Kapitel 7

## Zusammenfassung & Ausblick

In dieser Arbeit ist ein System zur modellbasierten Objekterkennung vorgestellt worden, dass die Modellaufbereitung automatisiert und auf den so erstellten Merkmalen eine Erkennung realisiert. Durch Implementierung des Konturkan-tenansatzes konnte sowohl die automatische Merkmalsgenerierung als auch die grundsätzliche Funktionalität einer Objekterkennung durch Projektionsoptimie-rung auf Basis der linearen Regression gezeigt werden. Die Möglichkeit der Er-mittlung einer Lagehypothese nur aus der Ansicht einer Kamera wurde ebenfalls vorgestellt. Eine Architektur zur Erweiterung auf weitere Identifikationsmerkmale wurde angelegt und die Möglichkeit der Verarbeitung zweidimensionaler Bildin-formationen mit dem modellbasierten, objektbezogenen Ansatz ist gegeben.

Die Erkennungsleistung des Systems ist zur Zeit noch als eingeschränkt zu be-werten, da mit der alleinigen Auswertung der Konturinformation eine robuste Erkennung beliebiger Objekte schwer zu realisieren, bzw. nur für ausgewählte Anwendungsbereiche geeignet ist. Die Auswertung auf Basis von künstlich er-zeugten Bildern kann die grundsätzliche Funktionalität der Algorithmen belegen, die Ergebnisse sind jedoch nur bedingt auf reale Einsatzfelder übertragbar, da durch einen inhomogenen Hintergrund und ungleichmäßige Beleuchtung deut-lich weniger ideale Bedingungen herrschen. Versuche unter realen Bedingungen im Rahmen einer konkreten Anwendung stehen somit noch aus. Dazu müssen Modelle existierender Objekte erstellt werden, mit denen die Modellaufbereitung

durchzuführen ist. Dies kann beispielsweise durch die Erfassung der entsprechenden Objekte durch 3D-Scanner oder durch die aufwändigere manuelle Eingabe der Objektgeometrie, realisiert werden. In diesem Zusammenhang ist zudem die Integration von Kameraansteuerungs- und -kalibrierungssoftware notwendig, um reale Szenarien aufnehmen zu können.

Ein weiteres zu lösendes Problem ist das der initialen Hypothese. In Kapitel 6.2.1 ist deutlich geworden, dass eine mehr oder minder grobe initiale Lageschätzung unbedingt notwendig für das Funktionieren der modellbasierten Erkennung durch den objektbezogenen Ansatz ist. Die Lösung dieser Problematik ist für Anwendungsgebiete, in denen kein Vorwissen ausgenutzt werden kann, sehr schwierig. Eine mögliche Lösung kann ein hybrider Ansatz aus ansichtsbezogener (vergleiche Abschnitt 2.2) und objektbezogener Erkennung sein, bei dem mehrere initiale Hypothesen geprüft werden.

Des Weiteren sollen die Möglichkeiten komplexerer Identifikationsmerkmale ausgeleuchtet werden. Der in dieser Arbeit gewählte Ansatz der Analyse der Konturkanten mit einem pixelbasierten Distanzmaß eliminiert sehr viele Informationen, die für eine Unterscheidung verschiedener Objekte wichtig sind. Die Erweiterung auf zusätzliche oder andere Merkmale, wie beispielsweise die angesprochenen Texturmerkmale, ist daher geboten. Eine weitere Möglichkeit sind Merkmale, die transformationsunabhängig sind. Dabei handelt es sich zumeist um Berechnungen von Größenverhältnissen geometrischer Aspekte, die auch bei Rotation und Projektion erhalten bleiben. Ein Beispiel hierfür sind die in [Jen94, Seite 126ff] beschriebenen *cross-ratios*.

Im Bereich der Modellaufbereitung, die natürlich auch für die Erweiterung auf neue Merkmale angepasst werden muss, ist besonders die Möglichkeit interessant, generische Modelle zu etablieren. Das bedeutet, dass, nach Auswertung mehrerer Modelle eines Objekttyps (z.B.: Couchtisch, Computertisch, Bürotisch, ...), die Aufbereitung eine Menge generischer Merkmale für die Objektklasse (hier: Tisch) erzeugt. Diese generischen Modelle sind nicht mehr, wie bisher angenommen, starr, sondern durch die Wahl geeigneter Parameter auf die sie etablierenden Prototypen abbildbar. Eine Erkennung erfolgt durch die Projektion dieser parametrisierten Modelle, so dass durch die vorgestellte globale Lageoptimierung ein

---

Rückschluss auf die wahrscheinlichste Objektklasse möglich ist. Darüberhinaus kann durch die lokale Optimierung – also den Vergleich einzelner Modellmerkmale mit den Bildmerkmalen basierend auf der Parametrisierung eine Klassifikation der Prototypen erfolgen. Dies ermöglicht dann auch eine Erkennung von Objekten, deren konkretes Modell nicht zur Verfügung steht.

In der vorliegenden Arbeit wurde die Erkennung auf Basis von Einzelbildern vorgestellt. Durch eine Erweiterung auf Bildsequenzen, ist die Möglichkeit gegeben, Hypothesen aktiv zu verifizieren. Ein mobiler Roboter oder Greifarm mit Kamera ist beispielsweise in der Lage ein Objekt aktiv von mehreren Seiten zu erkunden und so die Lagehypothese zu etablieren. Unter diesem Aspekt ist die Integration der Objekterkennung in ein Robotersteuerprogramm interessant, um dem System ein visuelles Explorieren von Szenen zu ermöglichen. Darüberhinaus ist auch die passive Analyse bewegter Szenen möglich, die die Eigenbewegung von Objekten nutzt, um zum einen durch Vorwissen eine initiale Hypothese zu erstellen (vergleiche Abschnitt 2.4), und zum anderen die Erkennung durch die sich ergebenden verschiedenen Perspektiven robuster zu machen.

Für eine vollständige Experimentierumgebung modellbasierter Objekterkennung sind zusätzliche Erweiterungen wünschenswert. So soll beispielsweise die Visualisierung von Modellen in das System integriert werden, und nicht wie bisher durch externe Programme wie **POV-Ray** realisiert werden.

Der Grundstein für ein System zur modellbasierten Objekterkennung ist also mit dieser Arbeit gelegt worden. Die Integration weiterer Komponenten ist möglich, um es zu einer kompletten Experimentierumgebung zur modellbasierten Objekterkennung auszubauen.





# Anhang A

## Einführung in die räumliche Geometrie

Vielen in dieser Arbeit verwendeten Algorithmen und Gleichungen liegen geometrische Beziehungen und Eigenschaften zu Grunde, welche an dieser Stelle kurz eingeführt werden sollen. Genauere Erklärungen zu den angeschnittenen Themen sind [Jän98] und [Jäh97] zu entnehmen.

### A.1 Homogene Koordinaten

Um die Transformationen, die bei der räumlichen Geometrie eine Rolle spielen, einfach formulieren zu können, werden die homogenen Koordinaten eingeführt. Hierzu wird einem dreidimensionalen Ortsvektor  $\vec{q}$  der die Lage eines Punktes bezüglich des Ursprungs angibt, eine weitere vierte Koordinate  $t$  hinzugefügt.

$$\vec{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} \quad \rightarrow \quad \vec{p} = \begin{pmatrix} tq_1 \\ tq_2 \\ tq_3 \\ t \end{pmatrix} \quad (\text{A.1})$$

Durch Division von  $\vec{p}$  durch  $t$  sind die dreidimensionalen Koordinaten aus den homogenen Koordinaten berechenbar. Im weiteren Verlauf wird  $t = 1$  angenommen, um die Rechnungen übersichtlich zu halten.

## A.2 Transformationsmatrizen

Ist ein Punkt  $\vec{p}$  in homogener Koordinatendarstellung gegeben, so können Transformationen dieses Punktes einfach in Form einer  $4 \times 4$ -Matrix dargestellt werden.

$$T = \begin{pmatrix} 1 & 0 & 0 & T_1 \\ 0 & 1 & 0 & T_2 \\ 0 & 0 & 1 & T_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Translation} \quad (\text{A.2})$$

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Rotation um die } R_x\text{-Achse}$$

$$R_y = \begin{pmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Rotation um die } R_y\text{-Achse}$$

$$R_z = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Rotation um die } R_z\text{-Achse} \quad (\text{A.3})$$

$$S = \begin{pmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Skalierung} \quad (\text{A.4})$$

Sollen mehrere Transformationen nacheinander durchgeführt werden, so ist dies durch Linksmultiplikation der entsprechenden Matrizen zu erreichen. Beispielsweise wird eine Rotation nach der *Roll-Pitch-Yaw*-Konvention durch die nacheinander erfolgenden Rotationen um die  $x$ -Achse, die  $y$ -Achse und schließlich die  $z$ -Achse realisiert. Eine nachfolgende Translation  $T$  kann zusätzlich mit eingerechnet werden:

$$R_{RPY} = T \cdot R_z \cdot R_y \cdot R_x$$

Diese Matrixmultiplikation stellt sich gemäß [FS] ausmultipliziert als

$$R_{RPY} = \begin{pmatrix} c_1 c_2 & -s_1 c_3 + c_1 s_2 s_3 & c_1 s_2 c_3 + s_1 s_3 & T_1 \\ s_1 c_2 & c_1 c_3 + s_1 s_2 s_3 & s_1 s_2 c_3 - c_1 s_3 & T_2 \\ -s_2 & c_2 s_3 & c_2 c_3 & T_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{A.5})$$

dar, wobei zu Gunsten einer lesbareren Formel folgende Substitutionen vorgenommen werden:

$$\begin{aligned} c_3 &= \cos \alpha, & s_3 &= \sin \alpha \\ c_2 &= \cos \beta, & s_2 &= \sin \beta \\ c_1 &= \cos \gamma, & s_1 &= \sin \gamma \end{aligned}$$

Durch Auswertung einer gegebenen Transformationsmatrix  $H$  ist es außerdem möglich, die Rotationswinkel zu errechnen, also den umgekehrten Weg zu beschreiten. Voraussetzung hierfür ist allerdings, dass die Matrix  $H$  tatsächlich eine Rotation (und Translation) ohne Scherungen und Verzerrungen beschreibt. Die Berechnung ist dann bis auf eine Singularität für  $\cos \beta = 0$  eindeutig durch

$$\begin{aligned} \alpha &= \Theta_y = \text{atan2}(H_{32}, H_{33}) \\ \beta &= \Theta_p = \text{atan2}\left(-H_{31}, \sqrt{1 - H_{31}^2}\right) \\ \gamma &= \Theta_r = \text{atan2}(H_{21}, H_{11}) \end{aligned} \quad (\text{A.6})$$

gegeben. Im Falle der Singularität kann nur eine Bedingung für  $\alpha$  und  $\gamma$  angegeben werden

$$\gamma - \alpha = \text{atan2}(-H_{12}, H_{22}), \quad (\text{A.7})$$

da in diesem Falle zwei Achsen zusammenfallen. Die Funktion  $\text{atan2}(y, x)$  errechnet den Arcustangens des Quotienten der Funktionsparameter, allerdings unter

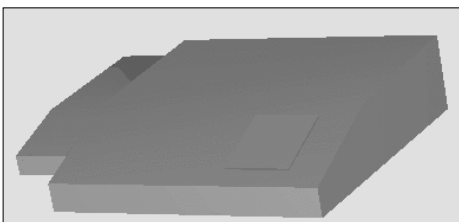
Berücksichtigung von Sonderfällen und Vorzeichen.

$$\text{atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & , \text{für } x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & , \text{für } x < 0, y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & , \text{für } x < 0, y < 0 \\ +\frac{\pi}{2} & , \text{für } x = 0, y > 0 \\ -\frac{\pi}{2} & , \text{für } x = 0, y < 0 \\ \text{undefiniert} & , \text{für } x = 0, y = 0 \end{cases} \quad (\text{A.8})$$

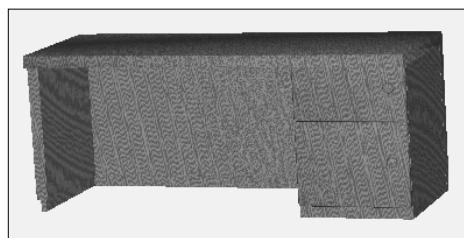
# Anhang B

## Verwendete Modelle

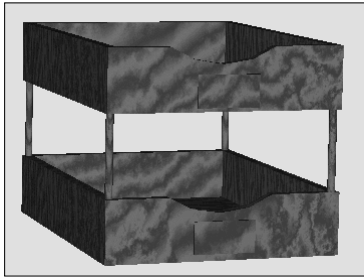
Die in der Arbeit verwendeten Modelle sind ausschließlich aus freien Quellen im Internet bezogen. Da die zu Grund liegenden realen Objekte nicht zur Verfügung stehen, werden die Algorithmen des Systems mit künstlichen Szenen getestet, welche mittels des Raytracing-Programms **POV-Ray** [pov01] erstellt werden. Zur Erzielung einer stärkeren Realitätsnähe sind die Objekte in den Untersuchungen mit objekttypischen Texturen belegt. Einige Beispiele der so erzeugten Modell sind hier dargestellt.



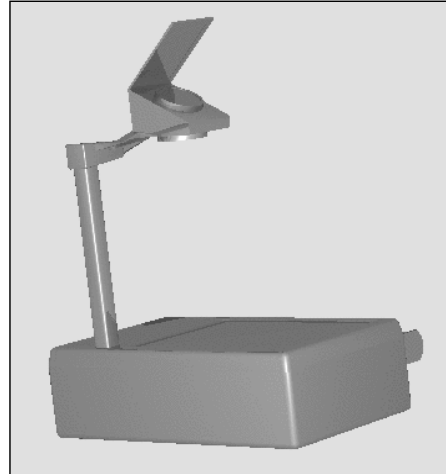
Ein einfaches Faxgerät



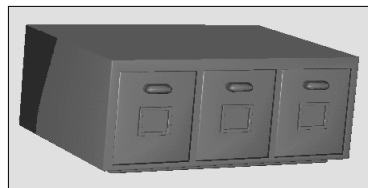
Ein Holzschreibtisch mit Schubladen



Eine Schreibtischablage aus Holz



Ein Overheadprojektor



Eine Diskettenbox aus Metall

# Literaturverzeichnis

- [Bai01] Thomas Baier. 3Dto3D Ein Programm zur Konvertierung diverser 3D-Formate, Juli 2001. <http://www.stmuc.com/thbaier/products.html>.
- [Can86] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8:679–698, 1986.
- [FS] E. Freund und O. Stern. Skriptum zur Vorlesung Robotertechnologie I. Institut für Roboterforschung, Dortmund.
- [HH94] D. Hausler und J.M. Herrmann. Laser triangulation: fundamental uncertainty in distance measurement. *Applied Optics*, 33:1306–1314, 1994.
- [Jäh97] Bernd Jähne. *Digitale Bildverarbeitung*. Springer Verlag, Berlin Heidelberg, 4. Auflage, 1997.
- [Jän98] Klaus Jänich. *Lineare Algebra*. Springer Verlag, Berlin Heidelberg, 1998.
- [Jen94] George A. Jennings. *Modern geometry with applications*. Springer Verlag, New York, 1994.
- [Kol93] Dieter Koller. Moving Object Recognition and Classification based on Recursive Shape Parameter Estimation. In *12th Israel Conference on Artificial Intelligence*, Seiten 359–368, Ramat Gan, Israel, 1993.
- [Kre98] Ulrich Krengel. *Einführung in die Wahrscheinlichkeitstheorie und Statistik*. Vieweg, 1998.

- [LZ96] S. Lanser und Ch. Zierl. MORAL: Ein System zur videobasierten 3D-Objekterkennung im Kontext autonomer, mobiler Systeme. In G. Schmidt und F. Freyberger, Hrsg., *Informatik aktuell: Autonome Mobile Systeme*, Seiten 88 – 98. Springer-Verlag, 1996.
- [MN00] K. Mehlhorn und S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, Februar 2000.
- [Pop94] Arthur R. Pope. Model-Based Object Recognition - A Survey of Recent Research. Technical report, The University of British Columbia - Department of Computer Science, 1994.
- [pov01] Persistence of Vision Ray-Tracer. <http://www.povray.org>, Juli 2001.
- [Rit96] Dieter Ritter. *Merkmalsorientierte Objekterkennung und Lokalisation im 3D-Raum aus einem einzelnen 2D-Grauwertbild und Referenzmodellvermessung mit optischen 3D-Sensoren*. Dissertation, Universität Nürnberg-Erlangen, 1996.
- [Sto93] Josef Stoer. *Numerische Mathematik I*. Springer Verlag, Berlin, 1993.
- [Wat89] Alan H. Watt. *Fundamentals of Three-Dimensional Computer-Graphics*. Addison-Wesley, Massachusetts, 1989.
- [Zel97] Andreas Zell. *Simulation neuronaler Netze*. Oldenbourg, München, 2. Auflage, 1997.